



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Prévision en temps réel de contraintes mécaniques en soudage par une bibliothèque de métamodèles

Real-time prediction of mechanical stresses in welding by a library of surrogate models

Soutenue par

Pablo PEREIRA ÁLVAREZ

Le 7 juillet 2022

École doctorale n°621

**Ingénierie des Systèmes,
Matériaux, Mécanique,
Énergétique (ISMME)**

Spécialité

Mécanique

Composition du jury :

Anthony GRAVOUIL Professeur, INSA Lyon	<i>Président</i>
Fehmi CIRAK Professeur, University of Cambridge	<i>Rapporteur</i>
Olga MULA Professeure assistante, TU Eindhoven	<i>Rapporteuse</i>
Pierre-Alain BOUCARD Professeur, ENS Paris Saclay	<i>Examineur</i>
Vincent ROBIN Ingénieur de recherche, EDF DT	<i>Examineur</i>
David RYCKELYNCK Professeur, Mines ParisTech	<i>Directeur de thèse</i>
Pierre KERFRIDEN Professeur assistant, Mines ParisTech	<i>Co-directeur de thèse</i>

Acknowledgements

Here I am, after three long years, giving the final closure to my PhD. At times, it seemed like an unsurmountable task but in the end it wasn't quite so thanks to many people that helped me and supported me, in a scientific and/or personal way. To all these people I dedicate the following paragraphs.

First of all, I would like to thank all the jury members. I am very glad that you all kindly accepted to participate in my PhD defense. In particular, I would like to thank Fehmi Cirak and Olga Mula's roles as rapporteurs. Your comments on my manuscript greatly encouraged me in the final weeks before the defense. Speaking about the defense, thank you as well to the examiners Anthony Gravouil and Pierre-Alain Boucard. The discussions that took place after the presentation were extremely interesting thanks to your interventions.

Allow me to switch into French to thank my Mines ParisTech and EDF supervisors. Merci beaucoup à mes deux directeurs de thèse, Pierre et David, pour votre soutien au cours de ces trois années. C'était un privilège de pouvoir travailler avec deux chercheurs de votre taille. Ensemble, vous faites une superbe équipe encadrante. J'ai toujours beaucoup appris en vous écoutant discuter. J'espère pouvoir collaborer avec vous dans le futur. À titre un peu plus personnel:

- David, merci beaucoup pour la confiance que tu as eu en moi dès notre première rencontre, quand j'étais stagiaire avec Tuan. J'ai découvert le monde passionnant de la réduction de modèle grâce à toi, mais ce que je voudrais remercier c'est la gentillesse, la bienveillance et la capacité d'écoute que tu as toujours eu envers moi.
- Pierre, merci pour tout ce que tu m'as appris et pour la passion que tu transmets pour la recherche. Depuis les premières fois où tu venais dans le bureau de David pour discuter, en passant par tous les jours où on a travaillé ensemble, c'était toujours un plaisir de travailler avec toi. Et je vais toujours me souvenir que grâce à toi je sais que je suis allergique aux racines carrées.

Je voudrais remercier maintenant Vincent, qui m'a beaucoup aidé sur les aspects expérimentaux de la thèse. Sans toi, je n'aurais pas pu conduire une campagne expérimentale comme on l'a fait. Merci aussi à Charles, qui m'a aidé avec les aspects mathématiques au début de la thèse, avant qu'il ne parte se rapprocher des vagues de l'océan Atlantique. Ces discussions ont continué avec David, arrivé à la fin, mais fondamental pour me soutenir pendant la rédaction. Merci beaucoup pour ton implication dans mon projet de thèse.

Il est difficile de dire combien d'encadrants j'ai eu à EDF car beaucoup de personnes ont été impliquées dans mon travail. Sans doute, même si officiellement ce n'était pas le cas, il y a deux personnes que je voudrais ajouter à cette liste d'encadrants. Merci Josselin et Sofiane, vous avez toujours tout donné pour moi

depuis mon premier jour à EDF, quand j'étais un jeune sorti d'une fac de maths qui ne connaissait rien au soudage. Que ce soit d'un point de vue personnel ou scientifique, vous avez toujours été là pour m'aider. Je vous remercie de tous vos mots d'encouragement pendant les années qu'on a passées ensemble.

Je ne peux pas oublier tous mes collègues à EDF qui ont fait de ces trois années une expérience très intéressante et enrichissante. Les séminaires, les discussions en salle de convivialité et les soirées parisiennes. Merci Claire, Julien, Laura, Louis, Merlin, Pauline, Sanaa, Thibault, Vanessa et Vincent. Merci surtout à tous mes collègues doctorants avec qui j'ai partagé ces années et plein de bons moments. Álvaro, fue un placer conocerte y pasar tanto tiempo contigo. Nada hubiera sido igual sin nuestras charlas que duraban horas y horas. Mala idea subir a tu despacho si queríamos ser productivos. Fermer la porte du bureau de Jérôme annonçait un bon moment que ce soit pour rigoler ou râler ensemble. Pareil quand j'entendais le classique 'Hola cabrón' d'Elias. Avec les autres membres du bureau de la chance, Sami et Théo, on ne savait jamais ce qui pouvait arriver, mais qu'est-ce qu'on rigolait bien. Une petite pause l'après-midi, Paul nous raconte ses dernières aventures dans le rouge. Antoine, Azénor, Clément, Marouane, Thomas... quelle génération de doctorants ! C'était un plaisir de passer du temps avec vous tous.

Je n'étais pas le seul à vivre ma thèse et j'en ai vécues deux autres pendant ces années. Noor et Virgile, mes amis qui ont été à côté de moi depuis si longtemps. Merci de m'avoir soutenu dès le début et j'espère avoir pu vous aider autant que vous le méritiez. Cécile, tu n'es arrivée qu'à la toute fin, quand je ne faisais qu'écrire, mais tu sais bien que tu m'as beaucoup aidé. Ça tombe bien, tu ne feras qu'écrire pendant ta thèse, et je serai toujours là pour toi. I per terminar, moltes gràcies també per a la Laura, que sempre va estar al meu costat. Malgrat la distància, vull que sàpigues que la teva ajuda va ser indispensable. No ho oblidaré mai.

¡Casi se me olvida! Gracias a mi familia, sobre todo a mis abuelos Carmen y Paco por quererme tanto. Si de verdad hay que darle las gracias a alguien, debe ser sin duda a mis padres, Juan y Marta, y a mi hermano Jaime. Muchas gracias a los 3 por haberme acompañado desde el principio (en este caso, no se puede ser más literal). Gracias, papá y mamá, por haberme dado la posibilidad y la motivación para conseguir todo lo que he hecho desde pequeño. Y a ti también, Jaime, el mejor compañero de piso y amigo que se pueda imaginar.

Contents

1	Introduction	1
1.1	Welding in the nuclear industry	3
1.2	Objectives	4
1.3	State of the art	5
1.3.1	Model Order Reduction	6
1.3.2	Inverse Problems	13
1.3.3	Classification and clustering problems in machine learning	18
1.4	Contributions	23
1.5	Computational tools	24
2	PVR hot-cracking tests: experiments and digital twining	25
2.1	The PVR experiment	27
2.1.1	The PVR experiment: a hot cracking test	27
2.1.2	The PVR specimen	28
2.1.3	Experimental configuration	30
2.1.4	Instrumentation	31
2.2	PVR experimental campaign	34
2.2.1	2019 campaign	34
2.2.2	2021 campaign	36
2.3	Digital twin of a PVR experiment	38
2.3.1	Thermomechanical model	38
2.3.2	Design of experiment	42
2.3.3	Calibration of Goldak's model	44
2.4	Parametric study of a PVR experiment	49
2.4.1	Design of Experiment	49
2.4.2	Simulation	49
2.4.3	Results	50
2.4.4	Dimension and size of the generated data	52
2.5	Merging experimental and simulation data	53
2.5.1	Estimating the infrared camera measurement error	53
2.5.2	Projection of the infrared camera data into the FE mesh	54
2.6	Conclusion	57
3	Local Probabilistic PCA surrogate models	59
3.1	Truth inverse problem	62
3.1.1	Parametrised probabilistic setting	62
3.1.2	"Truth" online Bayesian conditioning	63
3.2	Local multiphysics PCA model	64
3.2.1	Local surrogate models	65
3.2.2	Maximum likelihood PCA	67

3.2.3	Offline stage methodology	68
3.3	Online prediction in real time	68
3.3.1	Fast computation of mean and covariance	69
3.3.2	Known position	71
3.3.3	Unknown position	72
3.3.4	Online stage methodology	73
3.4	Forecasting of future states	74
3.4.1	Forecasting methodology	74
3.5	Numerical results	75
3.5.1	Estimation of the heat source position	75
3.5.2	Tests with noisy synthetic data	77
3.5.3	Tests with real experimental data	79
3.6	Conclusion	85
4	Clustering local reduced bases	87
4.1	Generalising the construction of local models	89
4.2	Clustering the simulation data	90
4.2.1	The K-medoids method	90
4.2.2	Description of the simulation data	92
4.2.3	Euclidean distance	92
4.2.4	Grassmann distance	93
4.2.5	Projection error in the state space	94
4.3	Offline stage results	96
4.3.1	Euclidean clustering	98
4.3.2	Grassmann clustering	99
4.3.3	Projection error clustering	101
4.3.4	Offline stage results recap	102
4.4	Online stage	103
4.4.1	Clustering labels	103
4.4.2	Optimal relabelling	105
4.4.3	Multi Layer Perceptrons	106
4.4.4	Training a classification Multi Layer Perceptron	109
4.5	Numerical results	110
4.5.1	Training of the MLP	111
4.5.2	3D temperature and stress estimation with noisy synthetic data	117
4.6	Conclusion	128
5	Multipass welding hyper-reduction	131
5.1	Introduction	133
5.2	Simplified multipass welding test case	134
5.3	Physical mechanisms captured by dynamic mode decomposition	134
5.4	Physical fields shift method	137
5.5	Local-global extrapolation method	138
5.6	Inherent strain method	139
5.7	Numerical results	140
5.7.1	Displacement and stress extrapolation	140
5.7.2	Plastic strain extrapolation	142
5.7.3	Hyper-reduced models from extrapolated fields	144

5.7.4 Execution time	147
5.8 Conclusion	148
6 Conclusion and perspectives	149
A Appendix A: Projection error algorithm	157
Bibliography	161

List of Figures

1.1	Automatised Tungsten Inert Gas (TIG) welding experiment at EDF R&D's welding lab.	3
1.2	a) Gappy data. b) Reconstruction of the face with 50 POD modes. c) Original face. d) Projection of the original face onto 50 POD modes. (Image taken from [151].)	17
1.3	General scheme of a classification problem.	18
2.1	Scheme of a PVR experiment.	27
2.2	Experimental configuration of a PVR test with two thermocouples.	28
2.3	Dimensions of the PVR specimen and position of the thermocouple.	30
2.4	Machines needed for a PVR experiments. On the left, TIG welding torch attached to a robotic arm. On the right, tensile loading machine.	31
2.5	Thermocouple measurement error. The use case for PVR experiments is circled in red.	32
2.6	Two stereo correlation cameras (red) on top of the infrared camera (blue).	34
2.7	Painted sacrificial specimen after welding.	35
2.8	Welded PVR specimen after the 2019 experiment.	36
2.9	Infrared camera image taken during the 2019 experiment projected onto the FE mesh.	36
2.10	Half-painted low energy specimen after welding.	37
2.11	Interactions between temperature, metallurgy and mechanics in a welding process.	39
2.12	Goldak's double ellipsoid model.	42
2.13	PVR mesh.	42
2.14	Temperature values at the thermocouple position for 400 samples of θ	47
2.15	Explained variance per number of modes in SVD.	48
2.16	Thermocouple and surrogate model result for the optimal θ	48
2.17	Snapshot of a temperature calibrated simulation of the 2019 experiment.	51
2.18	Snapshot of a principal stress field of a calibrated simulation of the 2019 experiment.	51
2.19	Cross-section view of the molten zone of a temperature snapshot of the 2019 experiment.	52
2.20	Force/displacement simulation and experimental profiles of the 3 energy levels.	52
2.21	Temperature measures taken with a thermocouple and an infrared camera obtained from the 2019 experiment.	53
2.22	Difference between camera and thermocouple data over time.	54
2.23	Raw infrared video frame. Colours represent Digital Levels.	55

2.24	Temperature values on the intermediate mesh. Example obtained from specimen 9 of the 2021 campaign.	56
2.25	Superposition of the intermediate and the PVR meshes. Example obtained from specimen 9 of the 2021 campaign.	56
2.26	Infrared temperature values on the PVR mesh. Example obtained from specimen 9 of the 2021 campaign.	57
3.1	Projection of infrared images on the FE mesh.	76
3.2	Possible PPCA bases with associated probability.	76
3.3	Noisy simulation data on camera area.	77
3.4	Temperature estimations, FE simulation and confidence interval. (a) Camera side (observed noisy FE data). (b) Torch side (No observed data).	78
3.5	Maximum principal stress estimations, FE simulation and confidence interval. (a) Camera side (observed noisy FE data). (b) Torch side (No observed data).	78
3.6	Experimental data - Temperatures above 400 °C	79
3.7	Temperature estimations and camera data.	81
3.8	Temperature estimations, FE simulation and confidence interval. (a) Camera side. (b) Torch side.	81
3.9	Maximum principal stress estimation, FE simulation and confidence interval. (a) Camera side (observed infrared camera data). (b) Torch side (no observed data).	81
3.10	Camera data of the future state.	82
3.11	Forecasted temperature and camera data.	83
3.12	Forecasted temperature, FE simulation and confidence interval. (a) Camera side. (b) Torch side.	83
3.13	Forecasted maximum principal stress, FE simulation and confidence interval. (a) Camera side. (b) Torch side.	83
3.14	Forecasted temperature conditioning by the known parameters (μ) and by the known and unknown parameters (μ, θ) with their correspondent confidence interval. (a) Compared to experimental camera data. (b) Compared to FE results.	84
3.15	Forecasted temperature conditioning by the known and unknown parameters (μ, θ) and posterior estimation using the infrared camera data with their correspondent confidence interval. (a) Compared to experimental camera data. (b) Compared to FE results.	85
4.1	Distribution of cluster sizes for each dissimilarity metric.	98
4.2	Distribution of speed, power and position for an arbitrary cluster (Euclidean distance).	99
4.3	Distribution of speed, power and position for an arbitrary cluster (Grassmann distance).	100
4.4	Welding torch positions represented in the 6 largest clusters (Grassmann clustering).	101
4.5	Distribution of speed, power and position for the largest cluster (Projection error).	101

4.6	Welding torch positions represented in the 5 largest clusters (Projection error clustering).	102
4.7	Scheme of a one hidden layer MLP.	107
4.8	Optimal label boxplots showing the number of snapshots per label for each clustering metric.	110
4.9	Evolution of the learning rate following a cosine schedule with $T_{mult} = 2$, and $T_i = 1$ (left) and $T_i = 10$ (right).	112
4.10	Evolution of the accuracy over 1000 epochs. No warm restarts.	112
4.11	Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 1$.	113
4.12	Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 10$.	113
4.13	Evolution of the accuracy over 1000 epochs. No warm restarts.	114
4.14	Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 1$.	114
4.15	Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 10$.	115
4.16	Evolution of the accuracy over 1000 epochs. No warm restarts.	115
4.17	Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 1$.	116
4.18	Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 10$.	116
4.19	Best temperature estimation - Euclidean clustering.	121
4.20	Best result compared to the reference FE simulation - Euclidean clustering.	122
4.21	Best temperature estimation - Grassmann clustering.	122
4.22	Best result compared to the reference FE simulation - Grassmann clustering.	123
4.23	Best temperature estimation - Projection Error clustering.	123
4.24	Best result compared to the reference FE simulation - Projection Error clustering.	124
4.25	Worst temperature estimation - Euclidean clustering.	125
4.26	Reference snapshot - Euclidean clustering.	125
4.27	Worst temperature estimation - Grassmann clustering.	126
4.28	Reference snapshot - Grassmann clustering.	126
4.29	Worst temperature estimation - Projection Error clustering.	127
4.30	Reference snapshot - Projection Error clustering.	127
5.1	Simplified scheme of a N passes multipass welding.	134
5.2	Example of the extrapolation of the first component of the displacement by using the DMD with $n = 9$, on the left the extrapolated displacement at the end of pass #10, on the right the FE prediction.	137
5.3	Approximation of the u_X component of displacement field in an industrial Overlay case.	138
5.4	Comparing the final state of displacements (u_X component) after the 11 th pass. (From left to right: $\varepsilon = 5 \cdot 10^{-4}$, Ward, FE simulation).	141
5.5	Comparing the final state of stress (σ_{XX} component) after the 11 th pass. (From left to right: $\varepsilon = 5 \cdot 10^{-4}$, Ward, FE simulation).	142

5.6	Comparing the final state of stress (σ_{XX} component) after the 20 th pass. The plastic strain predictions were computed using the Ward clustering method. (From left to right: view of the RID of a hyper-reduced simulation, inherent strain, FE simulation).	147
A.1	Distribution of speed, power and position for the first cluster when initialising with the maximum L^2 norm.	158

List of Tables

2.1	Chemical composition of the 316L stainless steel specimen (in percentages except for boron, which is expressed in particles per million).	29
2.2	Temperature dependent material parameters of 316L stainless steel: thermal conductivity, volumetric heat capacity, thermal expansion coefficient and Young's modulus.	29
2.3	PVR experiment parameter values.	35
2.4	PVR experiment parameter values for each energy level.	37
2.5	Experiment labels, energy levels and total experiment time.	38
2.6	Temperature dependent parameters of the elasto-visco-plastic Chaboche model.	41
2.7	Minimum and maximum values for the PVR operational known parameters μ .	43
2.8	Minimum and maximum values for the unknown parameters θ , as determined by the experts.	43
2.9	Number of CPUs and average computation time of the simulations.	50
3.1	Computation time with both expressions with two different observation functions.	71
3.2	Posterior estimation of the unknown parameters θ .	79
3.3	Posterior estimation of the unknown parameters θ .	80
4.1	Interquartile range mean and variance of each scaled parameter (Euclidean clustering).	99
4.2	Interquartile range mean and variance of each scaled parameter (Grassmann clustering).	100
4.3	Interquartile range mean and variance of each scaled parameter (Projection error clustering).	102
4.4	Amount of labels that were changed when comparing the clustering labels and the optimal labels with respect to the PPCA reconstruction error.	106
4.5	Accuracy results for each MLP and all training strategies.	117
4.6	Average temperature estimation error.	118
4.7	Average principal stress estimation error.	119
4.8	Accuracy results for each MLP with noisy input.	119
4.9	Percentage of snapshots with an error below a certain threshold.	120
5.1	Simplified test case operational parameters.	134
5.2	Comparing different methods to predict the 11 th pass.	141
5.3	Comparing different methods to predict the 11 th pass' plastic strain.	142
5.4	Comparing different methods to predict the 20 th pass' plastic strain.	144

5.5	Hyper-reduced computations of the 11 th pass.	146
5.6	Hyper-reduced computations of the 11 th pass.	147

Chapter 1

Introduction

Abstract:

This chapter introduces the industrial context of the PhD and the main industrial objectives. It also covers the most important research topics involved in this PhD and identifies the relevant previous results and approaches that have inspired the current work.

Résumé :

Ce chapitre introduit le contexte industriel de la thèse et les principaux objectifs industriels. Il présente aussi les sujets de recherche explorés pendant la thèse ainsi que les résultats et approches qui ont inspiré ces travaux.

Contents

1.1	Welding in the nuclear industry	3
1.2	Objectives	4
1.3	State of the art	5
1.3.1	Model Order Reduction	6
1.3.2	Inverse Problems	13
1.3.3	Classification and clustering problems in machine learning	18
1.4	Contributions	23
1.5	Computational tools	24

1.1 Welding in the nuclear industry

This PhD was conceived as a CIFRE fellowship ("Convention Industrielle de Formation par la Recherche") between EDF R&D, Mines ParisTech and ANRT ("Association Nationale Recherche Technologie"). The challenges presented in this document correspond to industrial problems faced by engineers in the Research and Development teams at EDF. The main objectives of EDF R&D are to improve the safety of production facilities, optimise their operating time and increase their production and environmental performance. In particular, the PhD took place at a department called PRISME ("Performance, Risque Industriel, et Surveillance pour la Maintenance et l'Exploitation"). PRISME is a multidisciplinary department that integrates researchers in domains such as numerical simulation, statistics, probability or signal processing but also in experimental domains such as welding mechanical experiments and other topics related to risks and maintenance of mechanical components of EDF's power plants.

Welding is an omnipresent method in many industries as an assembly or a repair technique. EDF is responsible for the robustness of its installations' components and making its maintenance operations safe. Welding involves complex multiphysical phenomena that are still hard to understand and, thus, to control. Hence, it is subject to many types of defects.

The high safety standards of the nuclear industry demand an even higher quality control of welding operations. The quality is controlled *a posteriori* by non-destructive examinations after the procedure. In the event of non-conformity, the weld has to be repaired, thus delaying either the maintenance or the manufacturing operation with significant financial consequences.



FIGURE 1.1: Automatised Tungsten Inert Gas (TIG) welding experiment at EDF R&D's welding lab.

In that context, EDF's R&D has worked for years in developing models for numerical simulation of welding in order to help understand all the physical phenomena and achieve a higher control of maintenance and manufacturing operations. Numerical simulation of welding enables the thermo-metallo-mechanical behaviour during welding to be described and the origin of defects to be understood. However,

the complex non-linear aspect of a welding operation simulation makes the computational cost of a high-fidelity simulation impossible to use for real-time control. Numerical simulation of welding is, nonetheless, useful for post-operation analysis of the state of the mechanical components.

Nevertheless, the combination of simulation with sensor data extracted from the instrumented execution of welding operations is a promising strategy to detect and locate any possible defects. Such a strategy would allow the identification of not properly known welding process parameters and the estimation of hidden variables such as the stress tensor [173, 152]. With access to this kind of information, real-time detection and location of drifts along the welding process becomes possible, as well as the estimation of potential defects by assessing stresses in the weld. Subsequent targeted examinations could be carried out a posteriori for regions with an identified high defect risk. The implementation of real-time monitoring of the manufacturing and repair welding processes would consolidate the control of these procedures and reduce the repair rate and the number of post-process checks, thus contributing to the optimisation of the planning of maintenance and repair operations. Finally, it would also help the design engineering justifications by including an analysis of the stress state of the components, which are known to affect the fit for service of metal parts in pipes and pressure vessels.

Real-time control of welding operations is, thus, the next step towards a better understanding and control of welding procedures in order to ensure the high safety demands of the nuclear industry.

1.2 Objectives

In recent years, the welding industry has seen numerous efforts to include newer technologies and intelligent systems [167]. New developments in machine learning and deep learning allow detecting defects in welding from experimental data qualitatively [111, 68]. In [68], an X-Ray acquisition is performed once the operation is finished. In [111], a real-time detection is performed but only for the surface temperature field around the heat source. At the same time, the improved power of computers in the last decade has allowed the appearance of high fidelity welding numerical simulation models [60, 43, 95] aiming at predicting 3D transient temperature fields and various mechanical variables such as the Cauchy Stress tensor. Improving the computational times has led researchers to the utilisation of Reduced Order Models (ROM) in welding [38, 176, 98]. Uncertainty in model parameters and measures is addressed with well-established uncertainty quantification methods [58], often through the resolution of inverse problems [40]. At the heart of all these new approaches, one can identify two primary data sources: experimental and simulation data.

Physical systems are mathematically modelled via parametrised Partial Differential Equations (PDE). The resolution of the PDE is obtained with a numerical scheme such as Finite Elements (FE) [48, 14, 3] or Finite Volumes (FV) [59, 92]. Simulation data is obtained by running a computer code that solves these physical models. It gives access to different physical fields in an entire specimen, including not directly observable data, such as stress. The main drawback is that numerical

simulation performed by using Finite Elements Analysis is time-consuming, especially for non-linear multi-physics problems such as welding. It also depends on modelling hypotheses and parameters that are not well known and need calibration.

Experimental data is readily available in situ and does not need knowledge of complex physical systems and mathematical models. However, it is often noisy, and some data can only be partially observed or not observed at all in real time, such as stresses. Some examples of partial observations are punctual thermocouple measurements or surface-level temperature measures with an infrared camera.

This thesis's main objective is to develop a numerical strategy that allows merging these two data sources to obtain real-time estimations of non-observed or partially observed fields in a welding application. Experts could then analyse the estimated physical fields to inform decisions on the quality of a welding operation. The main challenges are dealing with the uncertainty of the experimental data and the numerical models in order to obtain meaningful estimations of the mechanical state as well as unknown welding parameters in real time.

Hence, the present thesis' objective lies in the domain of Data Assimilation and inverse problems, where only partial observations of the physical state of the weld will be available. The experimental data will be coupled with high-fidelity simulation data of very large dimension. Model Order Reduction will be used to overcome the curse of dimensionality and obtain real-time solutions with efficient local surrogate models. The construction and optimal selection of the local models will involve clustering and classification algorithms.

1.3 State of the art

The general case studied in this document involves parametrised PDEs, which can be described in the form of:

Find $u \in V$ such that:

$$f(u, \mu) = 0 \quad (1.1)$$

where f represents a system of partial differential equations with adequate boundary conditions, μ is a vector of parameters in a parametric space $\mathcal{P} \subset \mathbb{R}^p$, and V is a Hilbert space. These parameters may refer to physical quantities such as emissivity, the heat capacity or the elastic modulus, or modelling parameters such as source terms. The PDE is often the mathematical formulation of a physics problem. For elliptic equations, its weak formulation is given by:

Find $u \in V$ such that, $\forall v \in V$:

$$a(\mu; u(\mu), v) = l(\mu; v) \quad (1.2)$$

with $a(\cdot, \cdot)$ is a parametric bilinear function, for linear problems such as elasticity, and $l(\cdot)$ is a bounded parametric linear form. In this thesis, we restrict our attention to thermomechanical problems that are approximated by a sequence of elliptic solutions. Assuming that $a(\cdot, \cdot)$ is continuous and coercive and that $l(\cdot)$ is continuous, the existence and unicity of a solution to the PDE, for a given value of μ , is guaranteed by the Lax-Milgram theorem. The solution is given by a function $u \in V$ that depends on the parameters μ and a variable $x \in \Omega \subset \mathbb{R}^d$, in a domain Ω which may

represent space and time. The solution to this problem with the Finite Elements method is given by a function in a discrete subspace $V_{\mathcal{N}} \subset V$. In this discrete space, the problem reads:

Find $u_{\mathcal{N}} \in V_{\mathcal{N}}$ such that, $\forall v_{\mathcal{N}} \in V_{\mathcal{N}}$:

$$a(\mu; u_{\mathcal{N}}(\mu), v_{\mathcal{N}}) = l(\mu; v_{\mathcal{N}}) \quad (1.3)$$

which can be written as a linear system of equations given by the matrix expression:

$$\mathbf{A}_{\mathcal{N}}(\mu) \mathbf{u}_{\mathcal{N}}(\mu) = \mathbf{l}_{\mathcal{N}}(\mu) \quad (1.4)$$

where $\mathbf{u}_{\mathcal{N}}(\mu)$ are the coordinates of the solution $u_{\mathcal{N}}(\mu)$ with respect to a basis of $V_{\mathcal{N}}$.

In future sections, when referring to the solution of the PDE for a specific value of μ , $u(x, \mu)$ will denote the \mathcal{N} -dimensional vector obtained as a solution of the Finite Elements Method.

The following sections will introduce some basic and classical notions of Reduced Order Modelling, the resolution of inverse problems, and classification problems in machine learning. Details will be given for methods considered of particular interest for this work.

1.3.1 Model Order Reduction

Model Order Reduction (MOR) methods aim at approximating the solution of a high-fidelity parametrised problem by a low-dimensional representation in a low-dimension subspace of $V_{\mathcal{N}}$ [123, 5, 9, 135, 143, 83, 36]. This is done by identifying similarities in the solutions of a PDE. In a parametric PDE such as Equation (1.1), different parameter values may produce similar solutions. In time-dependent PDEs, the solution for different instants may be similar. The main goal of Reduced Order Models (ROM) is to reduce the computational cost of solving the PDEs by capturing these similarities.

Data driven ROM methods are usually separated into an offline and an online stage. In the offline stage, a reduced basis is constructed by considering a training set of data. This step is computationally expensive, as it requires the computation of a large number of simulations for a variety of parameters in order to create the training data set. The online stage involves using the reduced model as a fast surrogate of the high-fidelity model used for the original simulations.

The basic idea of MOR is that a solution $u(x, \mu)$ can be approximated by a linear combination of functions in a subspace of lower dimension N with respect to \mathcal{N} , the dimension of u :

$$u(x, \mu) \sim \sum_{i=1}^N \phi_i(x) \gamma_i(\mu) = \mathbf{\Phi} \boldsymbol{\gamma} \quad (1.5)$$

where $\mathbf{\Phi} = [\varphi_1, \dots, \varphi_N] \in \mathbb{R}^{\mathcal{N} \times N}$ is called a reduced basis and $\boldsymbol{\gamma} \in \mathbb{R}^N$ contains the coordinates in the reduced space.

In projection-based model order reduction, the objective of using a reduced basis is not only to obtain a reduced-dimension approximation of the solutions to a PDE but also to calculate future solutions reducing significantly the computation

time. The original PDE problem was solved in V_N , but with a reduced basis Φ , it can be solved in a reduced dimension space spanned by the orthonormal basis functions $\{\varphi_i\}_{i=1}^N$. By Galerkin projection[144], the linear system of equations in (1.4) becomes:

$$\Phi^T A_N(\mu) \Phi \gamma(\mu) = \Phi^T l_N(\mu) \quad (1.6)$$

A fast resolution of this system of equations relies on the existence of an affine dependence of a and l with the parameter μ . Indeed, this avoids the recalculation of $A_N(\mu)$ and $l_N(\mu)$ for each new parameter μ .

The following sections will describe several methods to construct reduced bases, such as the Proper Orthogonal Decomposition, the Probabilistic Principal Component Analysis and the Reduced Basis Method. Furthermore, the Hyper-reduction will be introduced as an attempt to reduce the computation times further by dealing with the numerical bottleneck that may appear in Galerkin-POD methods. The notion of local reduced bases that are valid in a limited region of the parameter domain is also introduced, being a key aspect for this PhD. The final section briefly introduces the Proper Generalised Decomposition, a ROM method based on the hypothesis of separability of all problem variables.

Proper Orthogonal Decomposition

Among the Reduced Order Modelling methods, the Proper Orthogonal Decomposition (POD)[12, 93, 172] stands out, as it has been used in several domains for many years. It can be traced back to Lumley[99] for the study of turbulent flows, and its snapshot version was introduced by Sirovich[150]. It is also known as the Karhunen–Loève transform or the Principal Component Analysis (PCA) in other domains, when considering spaces of finite dimension. Its close ties to the Singular Value Decomposition (SVD) have been studied by several authors, such as Volkwein[166].

The snapshot version of POD assumes that $\mathbf{U} = [u_1, u_2, \dots, u_M] \in \mathbb{R}^{N \times N}$ is a matrix of rank $d \leq \min\{N, N\}$ containing N parametric solutions of the EDP as a training set of data. This is called a snapshot matrix. The main idea of the POD is to find a set of orthonormal basis functions $\{\varphi_i\}_{i=1}^d$ in a subspace such that the solutions of the parametrised PDE in the columns of \mathbf{U} can be expressed as a linear combination of them:

$$\mathbf{U} = \Phi \mathbf{\Gamma} \quad (1.7)$$

where $\Phi = [\varphi_1, \dots, \varphi_d] \in \mathbb{R}^{N \times d}$ contains the basis functions and $\mathbf{\Gamma} = [\gamma^1, \dots, \gamma^d] \in \mathbb{R}^{d \times N}$ is a coefficient matrix.

For any column u_j , $j = 1, \dots, N$, its representation in the basis $\{\varphi_i\}_{i=1}^d$ is given by the linear combination of the basis functions, with the coefficients of the corresponding column γ^j of $\mathbf{\Gamma}$:

$$u_j = \sum_{i=1}^d \gamma_i^j \varphi_i \quad (1.8)$$

The dimensionality reduction is obtained by reducing the number of basis functions chosen, i.e. how many columns of Φ are taken. The l -rank approximation with

the POD basis is given by the linear combination of the first l basis functions. It is the optimal l -dimensional approximation in the sense of the Frobenius norm, as proved in [166]. The l -rank approximation \tilde{U}_l reads:

$$\mathbf{U} \sim \tilde{\mathbf{U}}_l = \mathbf{\Phi}_l \mathbf{\Gamma}_l \quad (1.9)$$

where $\mathbf{\Phi}_l = [\varphi_1, \dots, \varphi_l] \in \mathbb{R}^{\mathcal{N} \times l}$ only contains the first l basis functions, $\mathbf{\Gamma}_l = [\gamma^1, \dots, \gamma^l] \in \mathbb{R}^{l \times N}$ now only has l rows and $\tilde{\mathbf{U}}_l$ is the l -rank approximation, which is formed by individual approximations $\tilde{\mathbf{U}}_l = [\tilde{u}_1^l, \dots, \tilde{u}_M^l]$.

The POD basis of rank l can be determined by solving the following minimisation problem:

$$\begin{aligned} \min_{\psi_1, \dots, \psi_l \in \mathbb{R}^{\mathcal{N}}} \sum_{j=1}^M \|u_j - \tilde{u}_j\|^2 &= \sum_{j=1}^M \|u_j - \sum_{i=1}^l \gamma_i^j \psi_i\|^2 \\ \text{s.t. } \psi_i^T \psi_j &= \delta_{ij}, \quad i, j = 1, \dots, l \end{aligned} \quad (1.10)$$

where $\gamma_j = \mathbf{\Psi}^T u_j$ and $\mathbf{\Psi} = [\psi_1, \dots, \psi_l]$.

The solution to this minimisation problem can be obtained by a Singular Value Decomposition of the matrix \mathbf{U} [166]. The truncated SVD of \mathbf{U} is given by:

$$\mathbf{U} = \mathbf{V} \mathbf{\Sigma} \mathbf{W}^T \quad (1.11)$$

where $\mathbf{V} \in \mathbb{R}^{\mathcal{N} \times d}$, $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}^T \in \mathbb{R}^{d \times N}$. Each vector of the reduced basis $\{\varphi_i\}_{i=1}^d$ corresponds to the columns of \mathbf{V} . Furthermore, the matrix $\mathbf{\Gamma}$ is also deduced from the SVD and is $\mathbf{\Gamma} = \mathbf{\Sigma} \mathbf{W}^T$. Because of the orthogonality, $\mathbf{\Gamma}$ can also be expressed as:

$$\mathbf{\Gamma} = \mathbf{\Sigma} \mathbf{W}^T = (\mathbf{V}^T \mathbf{V}) \mathbf{\Sigma} \mathbf{W}^T = \mathbf{V}^T (\mathbf{V} \mathbf{\Sigma} \mathbf{W}^T) = \mathbf{V}^T \mathbf{U} = \mathbf{\Phi}^T \mathbf{U} \quad (1.12)$$

In statistics, the POD is known as PCA [23], and it is usually calculated as the eigendecomposition of the covariance matrix \mathbf{C} of \mathbf{U} . For centered data, the covariance matrix is calculated as $\mathbf{C} = \frac{1}{\mathcal{N}-1} \mathbf{U} \mathbf{U}^T \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, assuming that the mean of \mathbf{U} is 0. The eigendecomposition is given by:

$$\mathbf{C} = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^T \quad (1.13)$$

where \mathbf{P} is the matrix of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues λ_i , $i = 1, \dots, d$.

The relationship between PCA and SVD can be found again. Taking the SVD of \mathbf{U} :

$$\begin{aligned} \mathbf{C} &= \frac{1}{\mathcal{N}-1} \mathbf{U} \mathbf{U}^T = \frac{1}{\mathcal{N}-1} (\mathbf{V} \mathbf{\Sigma} \mathbf{W}^T) (\mathbf{V} \mathbf{\Sigma} \mathbf{W}^T)^T = \frac{1}{\mathcal{N}-1} \mathbf{V} \mathbf{\Sigma} \mathbf{W}^T \mathbf{W} \mathbf{\Sigma} \mathbf{V}^T \\ &= \mathbf{V} \frac{\mathbf{\Sigma}^2}{\mathcal{N}-1} \mathbf{V}^T = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^T \end{aligned} \quad (1.14)$$

and we can deduce, that the left singular values and the eigenvectors of the covariance matrix are the basis functions, whilst the eigenvectors and singular values σ_i are related by:

$$\lambda_i = \frac{\sigma_i^2}{\mathcal{N} - 1} \quad (1.15)$$

Probabilistic PCA

A special case of PCA is known as Probabilistic PCA[158, 23, 113]. Tipping and Bishop developed it in an attempt to redefine the PCA from a Gaussian latent variable model, thus giving a probability model to the observed data. This approach is closely related to factor analysis[19, 114] and is based on a linear latent variable model[22]. In the EDP context, the linear latent variable model can be interpreted as a low dimensional approximation of a set of solutions $\mathbf{U} \in \mathbb{R}^{\mathcal{N} \times N}$ by a linear operator $\mathbf{\Phi} \in \mathbb{R}^{\mathcal{N} \times l}$ and reduced coordinates $\boldsymbol{\gamma} \in \mathbb{R}^{l \times N}$ with two additional terms: \mathbf{m} that allows the model to have non-zero mean, and $\boldsymbol{\varepsilon}$ that accounts for the error of the approximation:

$$\mathbf{U} = \mathbf{\Phi}\boldsymbol{\gamma} + \mathbf{m} + \boldsymbol{\varepsilon} \quad (1.16)$$

Probabilistic PCA introduces two hypothesis:

1. $\boldsymbol{\varepsilon}$ is an isotropic multivariate Gaussian noise and follows the distribution $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_{\mathcal{N}})$ (where $\mathbf{I}_{\mathcal{N}}$ denotes the identity matrix of dimension \mathcal{N}),
2. $\boldsymbol{\gamma} \sim \mathcal{N}(0, \mathbf{I}_l)$.

The first hypothesis, in conjunction with equation (1.16), imply the following conditional probability distribution:

$$p(\mathbf{U}|\boldsymbol{\gamma}) = \mathcal{N}(\mathbf{\Phi}\boldsymbol{\gamma} + \mathbf{m}, \sigma^2 \mathbf{I}_{\mathcal{N}}) \quad (1.17)$$

and with the marginal distribution of $\boldsymbol{\gamma}$ being Gaussian and independent of $\boldsymbol{\varepsilon}$, the marginal distribution of \mathbf{U} is also Gaussian and defined by:

$$p(\mathbf{U}) = \mathcal{N}(\mathbf{m}, \mathbf{\Phi}\mathbf{\Phi}^T + \sigma^2 \mathbf{I}_{\mathcal{N}}) \quad (1.18)$$

The model parameters \mathbf{m} , $\mathbf{\Phi}$, and σ^2 can be computed by Maximum Likelihood. The log-likelihood function is given by

$$\log \mathcal{L} = -\frac{N}{2} (\log(2\pi) - \log(|\mathbf{C}|) - \text{tr}(\mathbf{C}^{-1}\mathbf{S})) \quad (1.19)$$

where $\mathbf{C} = \mathbf{\Phi}\mathbf{\Phi}^T + \sigma^2 \mathbf{I}_{\mathcal{N}}$ is the covariance matrix and $\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{u}_i - \mathbf{m})(\mathbf{u}_i - \mathbf{m})^T$ is the sample covariance matrix. It is proven by Tipping and Bishop that there is an explicit solution of the maximisation of the likelihood. The Maximum Likelihood estimators for each parameter are given by:

$$\mathbf{m} = \bar{\mathbf{U}} = \frac{1}{N} \sum_{i=1}^N \mathbf{u}_i \quad (1.20)$$

$$\Phi = \mathbf{V}_l \sqrt{(\Lambda_l - \sigma^2 \mathbf{I}_N)} \mathbf{R} \quad (1.21)$$

$$\sigma^2 = \frac{1}{N-l} \sum_{j=l+1}^N \lambda_j^2 \quad (1.22)$$

Thus, \mathbf{m} is the empirical mean of \mathbf{U} . Matrices \mathbf{V}_l and Λ_l are truncated matrices of the Singular Value Decomposition of $\mathbf{U} = \mathbf{V}\Lambda\mathbf{W}^T$, where only the l largest singular values and their associated left singular vectors are retained. Finally, the matrix \mathbf{R} is an orthogonal rotation matrix that, in practice, is usually set as the identity matrix. These estimators also give the error term ε the sense of a truncation error, determined by the square of the non-selected singular values.

An alternative Expectation Maximisation algorithm to obtain these parameters was also proposed by Tipping and Bishop, as well as by Roweis[134], who had independently introduced a very similar model without specifying the nature of the Gaussian noise ε , being simply defined by a general covariance matrix.

The reduced coordinates of a solution \mathbf{u}_j , $j = 1, \dots, N$, with the PPCA can be calculated as a projection into the reduced-dimension space by algebraic manipulation as with PCA:

$$\gamma^j = (\Phi^T \Phi)^{-1} \Phi^T (\mathbf{u}_j - \bar{\mathbf{U}}) \quad (1.23)$$

Reduced Basis Method

The Reduced Basis Method (RBM) is a method that has been proposed as an alternative to the POD method proposed by several authors [136, 123, 101, 71]. The main difference with the POD is that the basis is built sequentially by searching an optimal point in the parameter space \mathcal{P} in order to compute a training set of data.

From a basis Φ^k containing k basis functions, the iteration at step $k+1$ finds the parameter value μ_{k+1} that maximises the error between the high fidelity solution and its low-dimensional representation with Φ^k :

$$\mu_{k+1} = \arg \max_{\mu \in \mathcal{P}} \|u(x, \mu) - \Phi^k \gamma(\mu)\| \quad (1.24)$$

where $\|\cdot\|$ is an adequate norm.

Once the parameter μ_{k+1} is selected, the reduced basis is updated by including the high-fidelity solution $u(x, \mu_{k+1})$ to a snapshot matrix. Thus, if Φ^k was constructed from a snapshot matrix $\mathbf{U}^k = [u(x, \mu_1), \dots, u(x, \mu_k)]$, Φ^{k+1} is constructed from $\mathbf{U}^{k+1} = [u(x, \mu_1), \dots, u(x, \mu_{k+1})]$

An alternative solution is to include the vector $\Delta_k(\mu_{k+1}) = u(x, \mu) - \Phi^k \gamma(\mu)$ directly into the reduced basis after orthonormalisation by a Gram-Schmidt process[4].

The RBM has been used for different applications and types of PDEs [63, 137] and it has also been used for Data Assimilation applications in [104].

Hyper-reduction method

The resolution of the Galerkin ROM, represented by a linear system of equations such as in Equation (1.6), can be inefficient when there is no affine dependency with respect to the parameters μ . Evaluation of the non-linear terms in high dimension \mathcal{N} represents a bottleneck in the computation of a solution. This has been observed by several authors, giving birth to hyper-reduction methods or hyper-reduced order models (HROM) [140, 141, 53].

An affine decomposition can be achieved by applying the Empirical Interpolation Method (EIM) [18, 103] or its variations: its discrete version, the Discrete Empirical Interpolation Method (DEIM) [33, 34] or the Generalised Empirical Interpolation Method (GEIM) [105]. The EIM method is a greedy approach that aims at constructing an approximation space of dimension K spanned by a set of functions $\{q_k\}_{k=1}^K$ and a set of interpolation points $\mathcal{X}_K = \{x_1, \dots, x_K\}$ for a sufficiently regular parametric function $g(x, \mu) : \Omega \times M \rightarrow \mathbb{R}^d$, such that the interpolator g_K can be written as:

$$g_K(x, \mu) = \sum_{k=1}^K q_k(x) \beta_k(\mu) \quad (1.25)$$

respecting the K interpolation constraints:

$$g_K(x_k, \mu) = g(x_k, \mu) \quad (1.26)$$

where $k = 1, \dots, K$. For more details on the EIM algorithm, we refer to Barrault [18]. Its discrete version, the DEIM avoids the greedy algorithm by using the space spanned by the columns of a POD basis as the approximation space.

The interpolation points can be used to define a Reduced Integration Domain (RID) $\Omega_Z \subset \Omega$ [142]. The idea behind the creation of a RID is to compute the solution of the PDE exclusively on the mesh nodes associated with the interpolation points selected by DEIM. The solution is known only on a limited amount of points. In order to extend the solution to the rest of the domain, the Gappy POD method (introduced later on in this chapter) is used. The RID should be large enough so that the problem is well-posed. This redefinition of the domain necessitates a redefinition of the boundary conditions of the problem [141]. Some examples of the application of HROMs in welding applications can be found in [38], where robustness was improved for non-linear simulation and a moving framework was introduced for moving heat sources, or in [176], which saw inclusion of the strain field to improve the performance of the elasto-plastic models. More recently, it has been applied to additive manufacturing in [98]. The hyper-reduction method has also been used in other domains such as fluid dynamics [31, 16].

Local ROMs

The previous sections have introduced Reduced Order Models from a global perspective; the solutions are approximated on a single subspace spanned from global basis functions. However, for highly non-linear or transient problems, a better approach would be to approximate the solutions by subspaces spanned from adequate local

basis functions [7]. Thus, the parameter (or time) domain is divided into regions that a local reduced basis will individually represent.

Partitioning the domain can be done in several manners. Some physical problems are well-suited for a time or space decomposition where physical phenomena are local by nature [44, 46]. In parametric problems the partitioning can be performed on the parameter state [62] or on the state space [168, 6, 131, 41]. The partition can be performed according to different criteria, for example the k-means algorithm with the Euclidean distance in [168] or a dissimilarity metric based on the projection error between snapshots in [6]. In the present work, two partitioning principles are explored. In Chapter 3, the domain is separated by the positions of a moving welding torch, whilst in Chapter 4, three clustering techniques are used to partition the parameter space or the state space.

A new parametric solution is calculated using the local model associated with the new parameter's region [8]. The selection of a local basis may prove a difficult task, which will be tackled in this work by a Gaussian mixture for position-based local bases (Chapter 3) and a classification machine learning algorithm for state-based local bases (Chapter 4). An alternative to the selection of local bases is interpolation on the associated Grassmann manifold [112, 9, 5]. This interpolation approach is based on the notion of Grassmann manifolds [2]. A reduced basis $\Phi \in \mathbb{R}^{\mathcal{N} \times d}$ spans a subspace $\mathcal{S}_d \subset \mathbb{R}^{\mathcal{N}}$. This subspace belongs to $\mathcal{G}(d, \mathcal{N})$, which is a Grassmann manifold defined as the set of all subspaces of $\mathbb{R}^{\mathcal{N}}$ that are of dimension d . A distance can be defined on $\mathcal{G}(d, \mathcal{N})$ [175] such that the subspaces spanned from two reduced bases can be compared. If $\Phi, \Psi \in \mathcal{G}(d, \mathcal{N})$, then this distance is defined as:

$$\delta_G(\Phi, \Psi) = \sqrt{\sum_{n=1}^d \lambda_n^2} \quad (1.27)$$

where λ_n , $n = 1, \dots, d$ are the n^{th} principal angles between Φ and Ψ , which are computed from a SVD of $\Psi^T \Phi = V \Sigma W^T$:

$$\lambda_n = \cos^{-1}(\sigma_n) \quad (1.28)$$

with σ_n the n^{th} singular value. This distance is the geodesic distance on $\mathcal{G}(d, \mathcal{N})$ when considering the manifold as a Riemannian manifold [175].

In the present document, the Grassmann distance will be used as a dissimilarity metric for a clustering algorithm in the partitioning stage of the construction of local reduced bases in Chapter 4. Such a distance gave satisfactory results for local hyper-reduced model in uncertainty quantification for thermomechanical prediction of lifetime for turbine blades in [41] and also for harmfulness prediction of voids in welded seams in [142].

Proper Generalised Decomposition

Another type of ROM method is the Proper Generalised Decomposition (PGD), a technique based on the a priori approximation of a PDE solution by a separated representation of the parameters. Assuming that the variable \mathbf{x} represents space and time, $\mathbf{x} = (\underline{x}, t)$, the solution $u(\underline{x}, t, \mu)$ is expressed as:

$$u(\underline{x}, t, \mu) = \sum_{i=1}^N X_i(\underline{x}) T_i(t) \prod_{k=1}^p M_{ik}(\mu_k) \quad (1.29)$$

This method finds its roots in Ladevèze's work on the LATIN method [88]. For a brief review of the history of the PGD method, we refer to [36]. It has seen applications in different domains, including welding [30] and to accelerate the resolution of inverse problems [138].

1.3.2 Inverse Problems

Many mathematical problems are stated in the following manner: given an input, find the output. This is the case of the PDE presented in Equation (1.1), where the objective is to find the solution of the PDE for a given set of parameters μ . This is called a *direct*, or *forward*, problem. However, another type of problem exists when parts of the parameters μ are unknown and only an observation $y = h(u(x, \mu)) \in \mathbb{R}^{N_{obs}}$ of the solution is available via an observation operator h . These kinds of problems are known as *inverse* problems. Inverse problems can be further divided into two main groups:

1. State estimation problems: the objective is to find an approximation \tilde{u} of $u(x, \mu)$ from the observations y .
2. Parameter estimation problems: the objective is to find an approximation $\tilde{\mu}$ of the parameter μ such that the observation y comes from $u = u(x, \tilde{\mu})$.

In this document, the objective is to estimate both the state and unknown parameters of the model. The p parameters can be separated into a set of n_μ known parameters μ and n_θ unknown parameters θ :

$$\begin{pmatrix} \mu \\ \theta \end{pmatrix} \in \mathbb{R}^p \quad (1.30)$$

The field solution of the PDE becomes $u(x, \mu, \theta) \in \mathbb{R}^{\mathcal{N}}$. The objective is to estimate the true joint vector z :

$$z = \begin{pmatrix} u(x, \mu, \theta) \\ \theta \end{pmatrix} \in \mathbb{R}^{\mathcal{N}+p} \quad (1.31)$$

from an observation $y = h(z)$.

The main difficulty in this PhD is the highly non linear causal relation between θ and multiphysics variables u , involving both temperature and mechanical variables.

The resolution of an inverse problem is often a difficult task. The difficulty comes from a variety of sources. They are most often ill-posed problems because they do not necessarily have a unique solution. A problem is considered well-posed in the sense of Hadamard [64] when:

1. a solution exists,
2. the solution is unique,

3. and the solution's behaviour changes continuously with the initial conditions.

The ill-posedness of the problem often comes from the existence of several solutions for the available observations, which is sometimes known as identifiability of the solution [11]. Another source of ill-posedness is the nature of the observations. The observations can be partial and contain *noise*, i.e. they are perturbed and not exact. Thus, an observation y can be modelled by a functional h and an observation error term:

$$y = h(z) + \epsilon \quad (1.32)$$

The observation error is unknown, and it is usually modelled by a random variable with 0-mean, without bias. In the case of experimental data, the error is usually produced by the measurement instrument or by numerical error (truncation or rounding, for example). The noise is usually identified as a zero-mean Gaussian distribution with a certain covariance matrix C :

$$\epsilon \sim \mathcal{N}(0, C) \quad (1.33)$$

Perturbations in the input data produced by noise can cause large differences in the result. The amount of available data can also influence the ill-posedness of the problem. When there are not enough observations, i.e. $N_{obs} < \mathcal{N} + p$, the problem is considered underdetermined, leading to multiple possible solutions.

Classical approaches for the resolution of inverse problems

All the approaches to solving inverse problems must deal with their ill-posedness. Two classical approaches can be developed. The first approach, often called the deterministic or variational approach, relies on functional minimisation. The solution to the problem is found by minimising a certain functional J that aims at representing the discrepancy between the available observed data and the model:

$$\tilde{z} = \arg \min_{z \in \mathbb{R}^{\mathcal{N}+p}} J(y, z) \quad (1.34)$$

Determining J is the main task. Once J is fixed, the minimisation problem can be solved using standard optimisation methods [116]. A usual choice is the least-squares formulation given in Equation (1.35).

$$\tilde{z} = \arg \min_{z \in \mathbb{R}^{\mathcal{N}+p}} \frac{1}{2} \|y - h(z)\|_2^2 \quad (1.35)$$

In this type of approach there usually is a regularisation term. Tikhonov regularisation [165, 25] adds a regularisation term $\alpha R(z)$ to the least squares formulation in Equation (1.35):

$$\tilde{z} = \arg \min_{z \in \mathbb{R}^{\mathcal{N}+p}} \frac{1}{2} \|y - h(z)\|_2^2 + \alpha R(z) \quad (1.36)$$

The addition of a regularisation term can be interpreted as the addition of prior information on the solution and aims at improving the identifiability of the solution. The scalar value α is called the regularisation coefficient. Finding an optimal value

for α can be done, for example, using the L-method, described in [67]. The choice of $R(z)$ is problem dependent, but a usual choice is:

$$R(z) = \|z - \bar{z}\|_2^2 \quad (1.37)$$

where \bar{z} represents prior knowledge of z , for example, its mean.

The second classical framework is the Bayesian approach[42], where the problem is treated from a probabilistic point of view. In this context, y , z and ϵ are all treated as random variables. The joint probability distribution of y and z is given by $p(y, z)$. From this joint distribution, the marginal distributions are defined as:

$$\begin{aligned} p(y) &= \int_{\mathbb{R}^{N+p}} p(y, z) dz \\ p(z) &= \int_{\mathbb{R}^d} p(y, z) dy \end{aligned} \quad (1.38)$$

This approach is based on the notion of conditional probability. The goal is to compute the conditional density $p(z|y)$ and is defined by:

$$p(z|y) = \frac{p(y, z)}{p(y)} \quad (1.39)$$

The conditional distribution can be calculated by applying Bayes theorem, which states:

Theorem 1. *Let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ be two random variables and $p(a, b)$ their joint probability distribution. The conditional probability of a knowing b , $p(a|b)$, is:*

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)} \quad (1.40)$$

In this formulation of the Bayes theorem, $p(a|b)$ is known as the *posterior density*, $p(b|a)$ is known as the *likelihood* and $p(a)$ is called the *prior density*. Thus, in the Bayesian approach, the goal is to compute the posterior distribution of the unknown vector z conditioned by the observations y :

$$p(z|y) = \frac{p(y|z)p(z)}{p(y)} \quad (1.41)$$

The posterior density $p(z|y)$ is the probability of vector z knowing the observations (usually measurements). The prior distribution $p(z)$ represents a specific knowledge about z that was available before observing y . The sources of this knowledge are varied and include expertise, some physical property or a model. The likelihood $p(y|z)$ can be interpreted as the probability that the observation functional h produces the vector y from z . Finally, the term $p(y)$ comes from the observations and is constant, meaning that it serves as a normalisation term. This leads to the common notation

$$p(z|y) \propto p(y|z)p(z) \quad (1.42)$$

which means that the posterior distribution is proportional to the product of likelihood and prior.

In a general case where there are no hypotheses on the probability distribution of z and y , an explicit formulation of the posterior density does not exist. A usual approach is to sample the posterior distribution using Markov Chain Monte Carlo (MCMC) methods [130]. The objective of these methods is the simulation of direct draws from complex distributions, in this case, the posterior distribution. There are many different MCMC algorithms, some of the most common one being Metropolis-Hastings [35].

There are plenty of examples of the application of Bayesian inference with MCMC methods in the literature, for instance, in [73], where a fully Bayesian approach is used to calibrate a computer model that predicts the shape of a cylinder after imploding it with a high explosive charge. The main disadvantage of this approach is that MCMC methods need an extensive amount of samples to converge. A common issue with the resolution of inverse problems is the curse of dimensionality, the matrices needed for most numerical solutions are massive, and computation times are long. In particular, for MCMC methods, when the samples are extracted from an expensive computer code like a FEM solver, their application becomes impractical. There exist numerous attempts at using surrogate models to accelerate these methods [109, 108], including ROM methods such as the PGD [138] or the RBM [40].

An alternative to MCMC methods are the Approximate Bayesian Computation (ABC) methods [39, 106].

Gappy POD

Gappy POD, introduced by Everson and Sirovich [51] is a method that appeared as a way of reconstructing partial pixel observation of human faces. It is a great example of the use of reduced models for the resolution of an inverse problem. In a first article, Sirovich and Kirby [151] used POD to create low-dimensional representations of a set of black and white pictures of human faces. The level of gray of each image is determined by a function $f(x)$, where x is the position [151]. Then, a grey scale matrix can be computed with each element of the matrix corresponding to a pixel. The image F is finally defined as a vector of concatenated rows of the grey scale matrix. Using a POD basis constructed from a database of 115 male human faces, they were able to reconstruct a whole face (see Figure 1.2, b)) from a limited amount of pixels that is referred to as gappy data (see Figure 1.2, a)). However, a major drawback of the Gappy POD method is that it cannot deal with noisy observations contrary to the PPCA.

The Gappy POD method works as follows. Let $\Phi = [\varphi_1, \dots, \varphi_N] \in \mathbb{R}^{\mathcal{N} \times N}$ a POD basis of dimension N describing pictures with \mathcal{N} pixels. An individual picture F can be described as $F = \Phi\gamma$, where γ is an N -dimensional vector containing the reduced coordinates. A partial observation of the face, HF is obtained by a boolean operator $H \in \mathbb{R}^{D \times \mathcal{N}}$ that selects a limited amount of pixels $D \ll \mathcal{N}$. It follows that the observation HF can be written as

$$HF = H\Phi\gamma \quad (1.43)$$

where $H\Phi$ restricts the POD basis to the observed pixels.

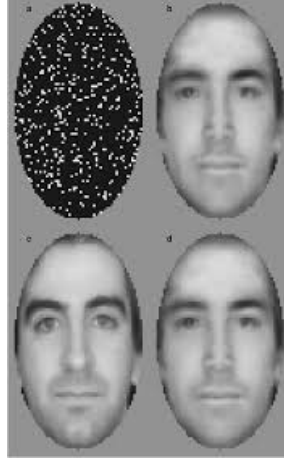


FIGURE 1.2: a) Gappy data. b) Reconstruction of the face with 50 POD modes. c) Original face. d) Projection of the original face onto 50 POD modes. (Image taken from [151].)

An approximation $\tilde{\gamma}$ of the reduced coordinates of the picture can be obtained from the limited observation by:

$$\tilde{\gamma} = ((H\Phi)^T H\Phi)^{-1} (H\Phi)^T (HF) \quad (1.44)$$

Thus, an approximation \tilde{F} of the original picture F is given by:

$$\tilde{F} = \Phi \tilde{\gamma} \quad (1.45)$$

Gappy POD has been used for many applications, for example, to approximate a whole field of unsteady flows from sensor measurements[171] or for the reconstruction of missing flow data[164]. It has also been used as a way of reconstructing a hyper-reduced simulation computed in a Reduced Integration Domain in [38, 45, 141].

Methods for state estimation in Data Assimilation

This final section on inverse problems briefly introduces some specific methods for the resolution of inverse problems. The first presented method is the Kalman Filter [79, 169], the most well-known data assimilation method. Filtering refers to the problem of updating the knowledge of a certain state or parameter from sequential observations [89]. Filtering usually involves two separate steps. First, a prediction of the future state using some linear mapping from the current state to a future one, and an update step where some data is observed and the prediction is corrected with this new knowledge. Kalman filters make some key assumptions. The first is that the prior and error models are Gaussian, allowing for an explicit expression of the posterior distribution. The second one is the markovian behaviour of the data sequences, where the current state depends exclusively on the previous state. A brief review of its industrial applications over the last decades is presented in [13].

Derivations of the Kalman filter exist to compensate for its shortcomings, most notably, the extended Kalman filter [129] that deals with non-linear dynamical systems, or the ensemble Kalman filter [50], which is designed to solve very large

problems by replacing the covariance matrix of the prior distribution by the empiric covariance of a state sample matrix.

Other notable methods are the 3D-Var and 4D-Var methods [11], two examples of the deterministic approach to inverse problems. Both methods aim at minimising a functional J (see Equation (1.34)) that represents the difference between all available data sources at a given instant, usually the previous estimation and observed data. The 4D-Var is an extension of 3D-Var used to estimate a trajectory of the system's state. This allows integration of the system's evolution over time as input data. 4D-Var is well-known for being the method used for weather forecasting [115]. In recent years, methods reminiscent of 3D-Var and 4D-Var have appeared, such as the Parametrised-Background Data-Weak methods [104, 66, 102] that include the use of ROMs for state estimation.

1.3.3 Classification and clustering problems in machine learning

Classification and clustering are, alongside regression and dimensionality reduction, the main type of problems in machine learning. In this document, they will be used to create and assign local reduced basis models. The construction of such models is motivated by the need to capture local physical phenomena during a welding operation.

Classification

Classification problems are *supervised* problems, meaning that their goal is to identify a certain mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ between inputs $x \in \mathcal{X}$ and outputs $y \in \mathcal{Y}$ [114]. In particular, in classification, the output space is a discrete set $\mathcal{Y} = \{1, \dots, K\}$ of dimension $K \in \mathbb{N}$, such that each value is a label designing a class. When there are only two classes, it is called binary classification.

Learning a classification model is done with a training dataset consisting of N pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. Figure 1.3 shows a basic scheme of the application of a classification algorithm to determine a predictor or classifier function.

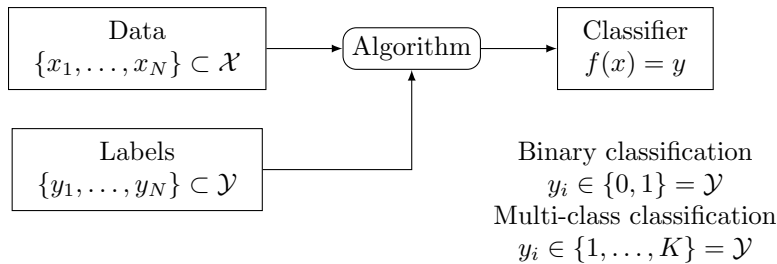


FIGURE 1.3: General scheme of a classification problem.

The main idea for the resolution of classification problems is to separate the input space \mathcal{X} into K *decision regions* by determining *decision boundaries* that allow assigning a label to each element of the input space. When the decision boundaries are linear, the methods that find them are called linear classification methods. There

exist many classification methods and algorithms. Bishop distinguishes three types of algorithms in [23]. The first type of model is called discriminant functions. Their objective is to construct a function that directly links each vector $x \in \mathcal{X}$ to a class. The other two types of models arise from a probabilistic view of the classification problem. The objective is to determine the conditional probability distribution $p(y = k|x)$, where $y = k$ indicates that x belong to the class C_k , which is the class associated to label $k \in \mathcal{Y}$. The two types of methods depend on how the conditional probability is calculated. Probabilistic generative models try to find $p(y = k|x)$ by applying Bayes' theorem (see Theorem (1)). Thus, they try to model the priors $p(y)$ and the likelihoods $p(x|y)$. The final type of methods are the Probabilistic discriminative models, which express $p(y = k|x)$ as a generalised linear model and determine its parameters by Maximum Likelihood.

Three classification methods are presented in the following section: the logistic regression method and classification neural networks. For a more detailed overview of these and other classification methods, we refer to one of the many books covering general topics in Machine Learning, such as [23, 114, 113, 69].

Logistic regression

Logistic regression is one of the classical methods for classification [23] and it is still important for its relationship to classification with neural networks, in particular with Multi Layer Perceptrons, the neural network architecture used in Chapter 4. Logistic regression can be divided into binary logistic regression and multinomial (or multiclass) logistic regression. The multinomial case assumes that there are K distinct classes. Let us also assume that $\{\mathbf{x}_i\}_{i=1}^N = \mathcal{X} \subset \mathbb{R}^d$, with $d \geq 1$, is a set of N samples and $\{y_i\}_{i=1}^N = \mathcal{Y} \subset \mathbb{R}$ is, likewise, a set of N label samples.

The conditional distribution $p(y|\mathbf{x})$ is supposed to follow a generalised Bernoulli distribution such that the probability of $\mathbf{x}_i \in \mathcal{X}$ belonging to each class C_k is given by:

$$p(y = k|\mathbf{x} = \mathbf{x}_i) = \frac{e^{(\boldsymbol{\alpha}_k^T \mathbf{x}_i + \beta_k)}}{\sum_{j=1}^K e^{(\boldsymbol{\alpha}_j^T \mathbf{x}_i + \beta_j)}} \quad (1.46)$$

where $\boldsymbol{\alpha}_j \in \mathbb{R}^d$, $j = 1, \dots, K$ are coefficient vectors of a linear model and $\beta_j \in \mathbb{R}$, $j = 1, \dots, K$ are the offset coefficients of a linear model. These coefficients are calculated by maximum likelihood estimation, which is computed by minimising the negative log-likelihood (NLL). The negative log-likelihood in this case is known as the cross-entropy error function. Let us introduce the notation $p_k(\mathbf{x}_i) = p(y = k|\mathbf{x} = \mathbf{x}_i)$. For N observations, the cross-entropy reads:

$$NLL = - \sum_{n=1}^N \sum_{k=1}^K \log p_k(\mathbf{x}_n) \quad (1.47)$$

The cross-entropy function is one of the usual choices for loss functions in classification neural networks.

Classification neural networks

Neural networks are non-linear statistical learning models that can be used for regression [154] and classification [127]. They can be applied to many types of data, for example: images with deep convolutional neural networks (CNN) [85], graphs with graph neural networks (GNN) [174] or sequences with recurrent neural networks (RNN) [110], which are used for language modelling. In recent years, there have been numerous developments in the deep learning field, including applications to the resolution of physical problems via Physics-informed neural networks [125], or to ROM with ROM-Nets [41]. This section introduces the general concept of neural networks for classification, and Chapter 4 presents the particular architecture of the Multi-Layer Perceptrons (MLP).

The main idea of neural networks is the successive feature transformation of the data along a certain number L of layers [114]. Mathematically, it is the composition of L non-linear differentiable functions $f_1(\cdot, \boldsymbol{\omega}_1), \dots, f_L(\cdot, \boldsymbol{\omega}_L)$:

$$f(\mathbf{x}, \boldsymbol{\omega}) = f_L(f_{L-1}(\dots(f_1(\mathbf{x}, \boldsymbol{\omega}_1))\dots, \boldsymbol{\omega}_{L-1}), \boldsymbol{\omega}_L) \quad (1.48)$$

where $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_L)$ are unknown coefficients that are learnt during the training process.

In the most simple neural network example, these coefficients refer to the coefficients of a linear combination of the previous layers' output features. This is the type of layer applied in MLPs and as the final layer of classification neural networks. Different types of layers can be used depending on the network architecture and type of data. For more types of layers, we refer to [113].

Defining \mathbf{z}_l as the output of layer l , if it is computed as a linear combination of the previous layer output, the current layer output reads:

$$\mathbf{z}_l = f_l(\mathbf{z}_{l-1}, \boldsymbol{\omega}_l) = \phi_l(\boldsymbol{\omega}_l^T \mathbf{z}_{l-1}) \quad (1.49)$$

where $\phi_l : \mathbb{R} \rightarrow \mathbb{R}$ is a non-linear differentiable activation function. This non-linear activation function is responsible for the non-linearity of the neural network models. Some examples of common activation functions are:

- ReLU function: $\phi_l(x) = \max(0, x)$
- Sigmoid function: $\phi_l(x) = \frac{1}{1+e^{-x}}$
- Hyperbolic tangent: $\phi_l(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

In a K-class classification task, the final layer usually applies a softmax function. The softmax function g is applied to a vector $\mathbf{x} \in \mathbb{R}^K$:

$$\begin{aligned} g : \mathbb{R}^K &\rightarrow \mathbb{R}^K \\ g(\mathbf{x}) &\rightarrow (g_1(x_1), \dots, g_K(x_K)) \end{aligned}$$

where each component g_k , $k = 1, \dots, K$ is computed as:

$$g_k(x_k) = \frac{e^{x_k}}{\sum_{i=1}^K e^{x_i}} \quad (1.50)$$

If vector \mathbf{x} comes from a linear combination in the previous equation, we recognise the similarity with the expression of the probability of an element belonging to a certain class in logistic regression, shown in Equation (1.46).

Training a neural network is done by a process called backpropagation, in which a particular loss function is minimised in order to calibrate ω . The gradient of f can be calculated with respect to the parameters of each function in the composition using the chain rule.

Clustering

Clustering, contrary to classification, is an unsupervised problem in which a group of N input points $x_i \in \mathcal{X}$ are separated into K classes, without any prior knowledge of the classes themselves. The data points are separated (or grouped) according to a certain dissimilarity metric δ :

$$\begin{aligned} \delta : \mathcal{X} \times \mathcal{X} &\longrightarrow \mathbb{R}^+ \\ (x, y) &\longrightarrow \delta(x, y) \end{aligned}$$

The goal is that all elements belonging to the same cluster are similar to each other and distinct from elements in other clusters [55]. The choice of metric is, thus, crucial to the clustering results and should be adapted to the data. An important distinction must be made between hard and fuzzy (or soft) clustering. In hard clustering, each data point belongs to only one class, whilst in fuzzy clustering, a data point can belong to one or more clusters with a certain probability. The output of a clustering algorithm is an assignment function f :

$$\begin{aligned} f : \mathcal{X} &\longrightarrow [0, 1]^K \\ f(x) &\longrightarrow (f_1(x), \dots, f_K(x)) \end{aligned}$$

where $\sum_{k=1}^K f_k(x) = 1$. In hard clustering, the image of \mathcal{X} by the assignment function is $\{0, 1\}^K$, meaning that the value of $f(x)$ is a vector $(0, \dots, 1, \dots, 0)$ with a 1 in the k^{th} position, which determines that x belongs to the class C_k .

There are two main types of clustering algorithms according to [23]: hierarchical algorithms and partitional algorithms. Hierarchical algorithms are further divided into agglomerative and divisive algorithms. Their main characteristics are summarised below:

- Agglomerative hierarchical algorithms start with N individual clusters, and, after each iteration, two clusters are merged.

- Divisive hierarchical algorithms start with one single cluster containing all the data points, and after each iteration, one cluster is divided into two clusters.
- Partitional algorithms create a non-overlapping partition of the data points. These methods include many different approaches, such as centre-based algorithms like k-means [100, 94], or density-based algorithms like DBSCAN [49, 149].

Similarly to the field of classification, there are numerous methods for clustering analysis. The following sections will present one of the most popular clustering methods, k-means. In Chapter 4, a divisive hierarchical algorithm, stemming from Amsellem's work on local bases [6], is presented. For a general overview of clustering, we refer to [55, 82] or to a general Machine Learning book that covers unsupervised learning [69, 114, 113].

K-means algorithm

K-means was first described by Macqueen [100] in 1967 and is based on the minimisation of the within-cluster sum of squares, which is the variance of the cluster and corresponds to the squared Euclidean distance between the elements of a cluster and its center, called mean or centroid. In k-means clustering, the number of clusters K is supposed to be known in advance.

Let us assume that \mathcal{X} is a subset of \mathbb{R}^d , with $d \geq 1$, and $\{x_i\}_{i=1}^N \subset \mathcal{X}$ is a set of data points. The squared Euclidean distance between two points $x, y \in \mathcal{X}$ is:

$$\delta_{EU}(x, y) = \sum_{i=1}^d (x_i - y_i)^2 \quad (1.51)$$

The within-cluster sum of squares is defined as:

$$w_k = \sum_{x \in C_k} \delta_{EU}(x, c_k) \quad (1.52)$$

where c_k is the centroid of cluster C_k . The centroid is calculated as the mean of all the elements in the cluster:

$$c_k = \frac{1}{\#C_k} \sum_{x \in C_k} x \quad (1.53)$$

The problem is, thus, to find the clusters C_1, \dots, C_K that minimise the within-cluster :

$$\arg \min_{C_1, \dots, C_K} \sum_{i=1}^K w_i = \arg \min_{C_1, \dots, C_K} \sum_{i=1}^K \sum_{x \in C_i} \delta_{EU}(x, c_i) \quad (1.54)$$

There are different algorithms and implementations that solve this problem, for instance the k-means++ algorithm presented in [10], which proposes a particular initialisation step. The most basic algorithm works as follows:

1. Choose an initial centroids set $C = \{c_1, \dots, c_k\}$

2. Assign a cluster to each element $x_i \in X$, $i = 1, \dots, N$ such that:

$$x_i \in C_k \iff \delta_{EU}(x_i, c_k) \leq \delta_{EU}(x_i, c_j) \quad \forall j \neq k \quad (1.55)$$

3. Recalculate all the centroids in C as the mean of elements in each cluster:

$$c_k = \frac{1}{\#C_k} \sum_{x \in C_k} x \quad (1.56)$$

4. Repeat steps 2 and 3 until the centroids in C don't change between two iterations.

The result of k-means clustering depends on the algorithm's initialisation, which is usually random. The initialisation is subject to study, as shown in [119, 32]. The other difficulty is the choice of K , which can be addressed, for example, with the silhouette score method [132, 82].

K-medoids

K-medoids is a variant of k-means where the centroid points, now called medoids, are chosen from the data set instead of being calculated as the mean of the elements in each of the K clusters. As for k-means, the number of clusters K is supposed to be known in advance. These medoids minimise the average dissimilarity with respect to the other elements of its cluster:

$$m_k = \arg \min_{x_i \in C_k} \frac{1}{\#C_k} \sum_{x_j \in C_k} \delta(x_i, x_j) \quad (1.57)$$

K-medoids main advantage is that it can be used with non-euclidean dissimilarity metrics. In fact, all the pairwise dissimilarities are pre-computed and stored in a $N \times N$ dissimilarity matrix D , where each component $D_{ij} = \delta(x_i, x_j)$, $i, j = 1, \dots, N$. For instance, a Grassmann distance between local reduced bases have been used as a dissimilarity measure in [142].

The classical algorithm is known as PAM (Partitioning around medoids), which was first proposed by Kaufman and Rousseeuw [133]. Other algorithms have been developed, most notably the CLARA and CLARANS algorithm, as well as their improved and faster versions [147]. A final algorithm, similar to the k-means, was proposed by Park et al. [117] and is claimed to be faster than the previous ones. This last algorithm, with an in-house implementation, will be used in Chapter 4.

1.4 Contributions

This section summarises the results produced during three years of work. It is divided as follows:

- Chapter 2 introduces the PVR experiment, which is selected as a welding operation application case of the methods presented in the following chapters. It is presented from an experimental and digital-twinning perspective, paying attention to the available data and how to treat it.

- Chapter 3 presents a position-based local multi-physics surrogate model that takes into account surface thermal data and transforms it into 3D estimations of temperature and stress. Details are given on the construction of the local model from multi-physics snapshot matrices. Numerical results are obtained from simulation and experimental input data, and the topic of future state forecasts is addressed.
- Chapter 4 generalises the creation of local models by presenting three clustering methods that aim at separating the domain optimally according to different metrics during the offline stage. The most adequate local model is assigned by a universal classifier obtained with a Multi-Layer Perceptrons neural network. The training process is discussed in detail, and numerical results with simulation data are shown.
- Chapter 5 presents an exploratory work aiming at producing reduced bases in the context of multipass welding. In particular, it deals with multipass welding cases with a large number of passes and presents two methods to extrapolate thermo-mechanical fields in order to construct reduced order models for welding beads that have not been simulated yet.

1.5 Computational tools

The high dimensionality of the problems tackled in the present work is directly responsible for memory allocation and computation time issues. Thus, it is essential to consider the computational tools used during the PhD. In terms of software, all the codes were homemade unless indicated otherwise (be it a python package or a simulation code). All the development was done in Python and aims at being as non-intrusive as possible.

In terms of hardware, two machines were used: a laptop and a scientific cluster, both of them using a Linux environment. The laptop contains an Intel(R) Core(TM) i7-6700HQ CPU 2.60GHz and 32 Gb of RAM. It has been used for numerical tests (without parallelisation) and part of the data post-processing (with parallelisation). The scientific cluster is situated in a distant datacenter and has a theoretical computational power of 2940 TFlops. It is composed of:

- 1112 standard nodes with 192 Gb of RAM per node, 36 cores per node and Skylake 6140 processors.
- 70 high memory nodes with 384 Gb of RAM per node, 36 cores per node and Skylake 6140 procesors.
- 32 graphical nodes equipped with 192 Gb of RAM per node and 2 NVIDIA Volta V100 GPUs per node.

It was mainly used for the Finite Elements simulations, the training of neural networks on the graphical nodes and the rest of the post-processing that needed to be applied to very large files.

Chapter 2

PVR hot-cracking tests: experiments and digital twining

Abstract:

This chapter will present the PVR experiment, which is the application case used in the following chapters to illustrate the methods developed in the present document. Here, we will describe a PVR hot cracking test. We will show the experiences carried out at EDF R&D welding lab, the type of available experimental data, and how to use them. Next, we will tackle the numerical simulation of these experiments, focusing on the calibration of the models thanks to the experimental data. Finally, we will show how to perform an exhaustive parametrical study of a PVR experiment. This study will be used as a numerical database in future chapters to build reduced-order models.

Résumé :

Ce chapitre présente l'essai expérimental PVR, qui est le cas test expérimental utilisé dans les chapitres suivants pour illustrer les méthodes développées dans ce manuscrit. Ici, on décrira un essai de fissuration à chaud PVR. On montrera les essais qui ont été réalisés au laboratoire de soudage d'EDF R&D, les différentes sources de données expérimentales et comment les utiliser. Après, on expliquera comment réaliser une simulation numérique de ce type d'essai, en se concentrant particulièrement sur la calibration du modèle à l'aide des données de thermocouples. Finalement, une étude paramétrique complète sera réalisée. Cette étude sera utilisée comme une base de données de simulation pour la construction de modèles réduits dans les chapitres suivants.

Contents

2.1	The PVR experiment	27
2.1.1	The PVR experiment: a hot cracking test	27
2.1.2	The PVR specimen	28
2.1.3	Experimental configuration	30
2.1.4	Instrumentation	31
2.2	PVR experimental campaign	34
2.2.1	2019 campaign	34
2.2.2	2021 campaign	36
2.3	Digital twin of a PVR experiment	38
2.3.1	Thermomechanical model	38
2.3.2	Design of experiment	42
2.3.3	Calibration of Goldak's model	44
2.4	Parametric study of a PVR experiment	49
2.4.1	Design of Experiment	49
2.4.2	Simulation	49
2.4.3	Results	50
2.4.4	Dimension and size of the generated data	52
2.5	Merging experimental and simulation data	53
2.5.1	Estimating the infrared camera measurement error	53
2.5.2	Projection of the infrared camera data into the FE mesh	54
2.6	Conclusion	57

2.1 The PVR experiment

The use case for this PhD is the PVR hot-cracking test. It was chosen as an application case because it represents fairly well the conditions found in a real-life welding operation in EDF's nuclear power plants. In particular, it is the same kind of welding procedure (TIG welding) used in a production site, and the material of the specimen (316L stainless steel) is often used in the nuclear industry. This section will describe the hot-cracking test and the specimen. Then, the experimental configuration used in EDF R&D's welding lab is presented, including the instrumentation and a description of the measurement instruments.

2.1.1 The PVR experiment: a hot cracking test

The PVR, for Programmierter-Verformungs-Risstest (meaning cracking test with programmable deformation), was developed during the 70s by the Austrian company Boehler. It consists of a fusion line performed on a specimen being uniaxially tensile loaded [80, 124], as seen in the diagram in Figure 2.1. The deformation rate is augmented linearly to a predefined maximum value during the test. The main advantage of PVR tests is that they allow one to observe hot cracks of all types [80], including liquation and solidification cracks. These cracks appear in the heat-affected zone around the welded fusion line. The PVR test has been used to study the occurrence of hot cracking in different alloys [54].

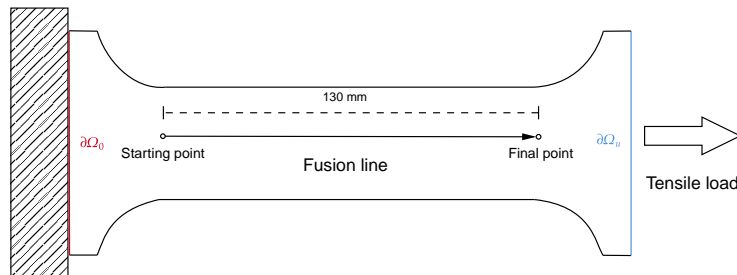


FIGURE 2.1: Scheme of a PVR experiment.

From a data assimilation perspective, the large surface of the PVR specimen allows easy instrumentation of the experiment in a lab environment. Some of the instrumentation options are:

- Placement of thermocouples for punctual temperature measurements.
- Filming with an infrared camera for surface-level temperature measurements.
- Mechanical force and displacement data given by the tensile load machine.
- Filming with stereo correlation cameras for strain and displacement data.

In Figure 2.2, one can observe an instrumented specimen before a PVR experiment. In this image, the specimen is instrumented with two thermocouples (see

the black wires on both sides of the specimen) and inserted into the tensile loading machine. A brief explanation of each instrument and the data they provide is given in the following sections.



FIGURE 2.2: Experimental configuration of a PVR test with two thermocouples.

2.1.2 The PVR specimen

The chosen specimen material is 316L stainless steel, an austenitic steel used regularly in the nuclear industry [17, 15]. The chemical composition of the material is summed up in Table 2.1. On Figure 2.2 one can notice that the design of the specimen borders fit clamping jaws of the tensile-loading machine.

The temperature-dependent material properties were obtained from a variety of results, including material characterisation by Geeble tests performed during Minh Le's PhD [90] and studies by CEA Saclay. They are summed up in Table 2.2. The parameters are not defined for all temperatures. Missing data points are linearly interpolated between available data. In absence of interpolation points, the last value is extended as a constant if needed.

TABLE 2.1: Chemical composition of the 316L stainless steel specimen (in percentages except for boron, which is expressed in particles per million).

B	C	Mn	Si	P	S	Cr	Ni	Mo	Co	Cu	Al	N
39 ± 4	0.016	1.59	0.540	0.027	0.0011	17.25	10.03	2.05	0.080	0.106	0.043	0.052

TABLE 2.2: Temperature dependent material parameters of 316L stainless steel: thermal conductivity, volumetric heat capacity, thermal expansion coefficient and Young's modulus.

Temperature ($^{\circ}\text{C}$)	$k (\frac{W}{mmK})$	$\rho c_p (\frac{J}{mm^3K})$	$\alpha (\frac{1}{K})$	E (MPa)
20	14×10^{-3}	3.784×10^{-3}	15.5×10^{-6}	190×10^3
100	15.2×10^{-3}	-	16×10^{-6}	-
200	16.6×10^{-3}	4.036×10^{-3}	16.6×10^{-6}	-
300	17.9×10^{-3}	-	17.1×10^{-6}	-
400	19×10^{-3}	4.302×10^{-3}	17.5×10^{-6}	-
500	20.6×10^{-3}	-	18×10^{-6}	-
600	21.8×10^{-3}	4.557×10^{-3}	18.4×10^{-6}	140×10^3
700	23.1×10^{-3}	-	18.7×10^{-6}	-
775	-	-	-	56.3×10^3
800	24.3×10^{-3}	4.823×10^{-3}	19×10^{-6}	-
850	-	-	-	56.3×10^3
900	26×10^{-3}	-	19.2×10^{-6}	-
1000	27.3×10^{-3}	5.072×10^{-3}	19.4×10^{-6}	-
1150	-	-	-	37.3×10^3
1200	29.9×10^{-3}	5.327×10^{-3}	-	-
1250	-	-	-	20.3×10^3
1384	-	5.572×10^{-3}	-	-
1390	-	7.823×10^{-3}	-	-
1394	-	11.175×10^{-3}	-	-
1400	32.5×10^{-3}	-	19.6×10^{-6}	-
1404	-	22.35×10^{-3}	-	-
1420	-	44.70×10^{-3}	-	-
1425	-	52.15×10^{-3}	-	-
1450	-	5.7×10^{-3}	-	-
1600	-	5.7×10^{-3}	19.7×10^{-6}	-
2400	32.5×10^{-2}	-	-	-

Previous studies at EDF R&D have used the same design but with different dimensions [159]. In that case, the specimen's dimensions were 260 mm long, 80 mm large and 3.5 mm thick. The ones used in this document are shown in Figure 2.3. They are 200 mm long, 80 mm large and 3 mm thick. The central area is where a fusion line is done, along a line of 130 mm . This fusion line aims at doing a welding operation easily reproducible. This part is 40 mm large.

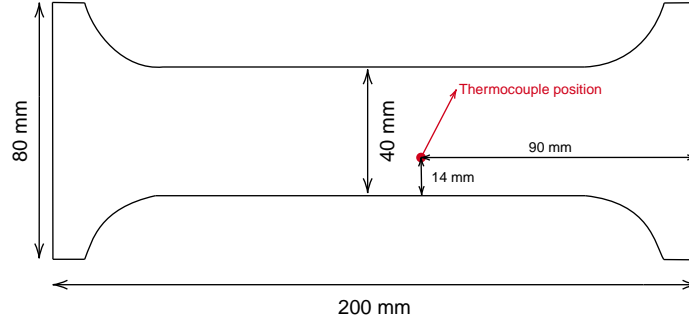


FIGURE 2.3: Dimensions of the PVR specimen and position of the thermocouple.

2.1.3 Experimental configuration

Two instruments are needed for a PVR experiment, a welding torch and a tensile loading machine. The fusion line is done with a ValkWelding TIG-welding torch attached to a 6-axes Panasonic robotic arm (see Figure 2.4, on the left). The heat source's velocity ($v\text{ [mm/s]}$), voltage ($U\text{ [V]}$) and current ($I\text{ [A]}$) are the three operational parameters that are fixed at the start of the experiment. The voltage and current are going to be represented together as the torch's power ($Q\text{ [W]}$) by the following relationship:

$$Q = U \cdot I \quad (2.1)$$

This choice is motivated by the equivalent heat source model [61], which is used to approximate the heat source during a welding operation. It also allows reducing the number of operational parameters to two. Using a robotic arm instead of a manual welder allows greater control of the operational parameters, including a constant torch velocity and a constant flux of shielding gas, which in this case is argon.

In regards to the tensile loading machine, a Lloyd Instruments LS100 plus (Figure 2.4, on the right), the maximum deformation rate, in mm/s is fixed in advance. The experiment starts with a constant force of 5000 N applied to the specimen and, as soon as the torch starts working, the deformation rate is augmented linearly up to the maximum deformation rate for the duration of the test.

One could also consider the duration of the experiment and the maximum deformation rate as two more operational parameters. The experiment duration is

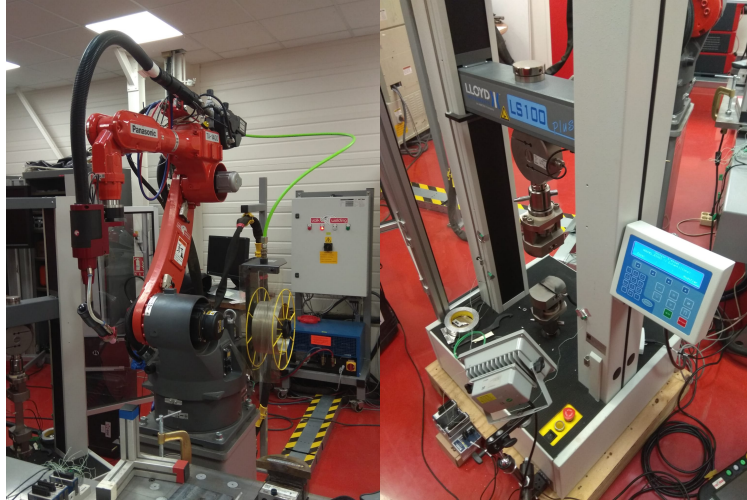


FIGURE 2.4: Machines needed for a PVR experiments. On the left, TIG welding torch attached to a robotic arm. On the right, tensile loading machine.

discarded as a parameter because it depends directly on the torch speed. In practice, all the experiments realised at EDF's welding lab used the same maximum deformation rate values to ensure repeatability for other engineering projects outside this research work. Thus, it is considered a constant operational parameter in our examples.

The successive steps of the PVR experiments are:

PVR experiment steps

1. The specimen is inserted into the tensile loading machine and is loaded with a constant force of 5000N.
2. The torch, located at the starting position, is initiated with the selected current and voltage.
3. The torch moves straight at the selected constant velocity while the specimen is deformed with a deformation rate that augments linearly.
4. The torch arrives at the ending position, it is turned off, and the tensile loading machine starts relaxing the deformation.

2.1.4 Instrumentation

As indicated previously, a great advantage of the PVR test is that it offers the possibility of further instrumentation to obtain real-time data. In particular, the experiments are instrumented with two homemade type-K thermocouples and filmed with an SC7500 FLIR infrared camera. The infrared video will provide real-time temperature measurements on part of the specimen's surface. Whilst both sides of the specimen have thermocouples, only the surface that is not being welded is filmed

to avoid reflections from the welding arc and the robotic arm. Additionally, some of the experiments were instrumented with stereo-correlation cameras.

Thermocouples

Thermocouples consist of two different electric conductors that measure an electric temperature-dependent voltage that can be interpreted as a temperature measurement. The thermocouples used at the welding lab are type K, which use two nickel alloys, chromel and alumel, as conductors.

The thermocouples are manually welded to the specimen with a ceramic cover to isolate them from the exterior. They are located close to the centre of the specimen as indicated in Figure 2.3. The thermocouples measure temperature in 0.02 seconds intervals, i.e., with a frequency of 50 *Hz*. The operating temperature range is between -200°C and 1370°C . However, they are more accurate between 0°C and 1100°C according to the calibration curves in Figure 2.5, which was provided by the manufacturer of our National Instruments NI 9213 thermocouple data acquisition module.

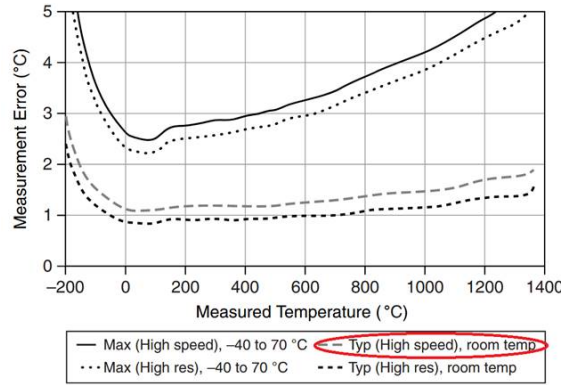


FIGURE 2.5: Thermocouple measurement error. The use case for PVR experiments is circled in red.

Infrared camera

An infrared camera is a device that produces images using light in the infrared range (wavelengths between $1\mu\text{m}$ and $14\mu\text{m}$). All objects with temperatures higher than the absolute zero emit thermal radiation in different frequencies. This spectral radiation is observable. Planck's law allows to relate the monochromatic spectral luminance (quantity of energy that is radiated) to temperature for a black body:

$$L(\lambda, T) = \frac{c_1}{\lambda^5} \frac{1}{\exp\left(\frac{c_2}{\lambda T}\right) - 1} \quad (2.2)$$

where $c_1 = 2hc^2$ and $c_2 = \frac{hc}{k}$ are the first and second radiation constants respectively. They are expressed in terms of three other physical constants: Planck's constant h , Boltzmann's constant k and the speed of light c .

For non-black bodies, the luminance is somewhat reduced by the emissivity of the material. The emissivity depends on the temperature, the wavelength and the

angle of observation θ . It is defined as the ratio between an object's luminance $L(\lambda, T, \theta)$ and a black body's luminance ($L^0(\lambda, T)$):

$$\varepsilon(\lambda, T, \theta) = \frac{L(\lambda, T, \theta)}{L^0(\lambda, T)} \quad (2.3)$$

If the emissivity of the observed material is known, Planck's law can be used to deduce the temperature of an object. It is usually complicated to determine the emissivity precisely. In order to avoid this problem, bichromatic systems observe a scene for two wavelengths λ_1 and λ_2 . This way, the temperature can be calculated, by inversion, from the ratio between the two luminances L_1 and L_2 :

$$\frac{L_1}{L_2} = \left(\frac{\lambda_1}{\lambda_2} \right)^{-5} \frac{\exp\left(\frac{c_2}{\lambda_2 T}\right) - 1}{\exp\left(\frac{c_1}{\lambda_1 T}\right) - 1} \quad (2.4)$$

The infrared camera at EDF's welding lab is an SC7500 FLIR infrared camera (see Figure 2.6, in blue). The resolution of the camera images is 320×256 pixels. This camera uses a filter wheel with 8 slots that are used simultaneously to observe a scene at different spectral bands, allowing to capture several temperature ranges. According to the calibration, it is able to identify temperatures between $-20^\circ C$ and $3000^\circ C$. The filters can be used by pairs while analysing a scene with two spectral bands. When all the filters are used, the maximum acquisition frequency is $383 Hz$. The maximum per filter frequency is divided by 8.

The camera films the surface that is not being welded in order to avoid reflections from the welding arc. The camera is fixed on top of a tripod and films a fixed area, i.e., the video does not follow the torch.

Stereo-correlation cameras

Digital Image Correlation (DIC) is a technique that measures displacement with a set of two cameras (see Figure 2.6, in red). The specimen needs to be painted in black with white drops randomly distributed on its surface so that the cameras can identify differences in the positions of the white drops. Figure 2.7 shows an example of a painted sacrificial specimen after a PVR experiment. The paint is temperature resistant up to $700^\circ C$. The white points position's differences are measured according to a calibration scheme. Two cameras are needed to observe depth and perform 3D analysis. The cameras used in the welding lab are two Pike F-421 with a resolution of 2048×2048 and Schneider-Kreuznach precision lenses. They operate with a maximum frame rate of $16 Hz$ at full resolution. In our experiments, the cameras capture thousands of images with a frequency of $10 Hz$. These images are later treated with specialised software such as Vic-3D or Vic-2D for 3D and 2D analysis, respectively.

The data offered by the stereo-correlation cameras have not been exploited in this document. Although the mechanical information obtained with these cameras could be of great interest, treating the thousands of images requires a long time. Exploration of such data and their use to validate the performances of a real-time deviation detection tool for welding operations is postponed to further research.



FIGURE 2.6: Two stereo correlation cameras (red) on top of the infrared camera (blue).

2.2 PVR experimental campaign

Two different experimental campaigns were carried out at EDF's welding lab. The first one, which was used for most of the numerical tests in this document, was done late in 2019 and involved a single PVR test. The second one, performed by the end of 2021, involved a total of 6 PVR tests. Unfortunately, these last experiments took place very late in the PhD due to Covid-related issues and could not be used for the numerical tests. The following sections will discuss both campaigns.

2.2.1 2019 campaign

The first experimental campaign aimed at reproducing the results of a previous PhD at EDF [160] with a different specimen geometry. As stated previously, the PVR specimen had larger dimensions in that work. The same operational parameters found in Giai Tran Van's thesis [160] were used for this experiment: a heat source speed of 2 mm/s , a current of 81 A and a voltage of 8.4 V . The maximum deformation rate is 20 mm/min , or 0.333 mm/s . All the parameters are summed up in Table 2.3.



FIGURE 2.7: Painted sacrificial specimen after welding.

TABLE 2.3: PVR experiment parameter values.

Current	81 <i>A</i>
Voltage	8.4 <i>V</i>
Travel speed	2.00 <i>mm/s</i>
Shielding gas	Argon
Maximal stroke rate	20 <i>mm/min</i>
Linear energy	340.2 <i>J/mm</i>

The experiment was instrumented with two type-K thermocouple positioned on both sides of the specimen as indicated in Figure 2.3 and an infrared camera. The camera films a portion of the specimen that is fixed during the entire experiment, illustrated in Figure 2.9.



FIGURE 2.8: Welded PVR specimen after the 2019 experiment.

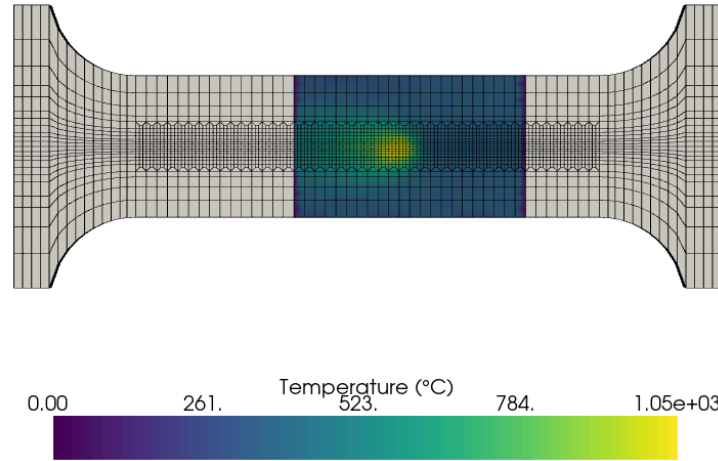


FIGURE 2.9: Infrared camera image taken during the 2019 experiment projected onto the FE mesh.

2.2.2 2021 campaign

The experiments in this campaign were divided into three sets according to the linear energy transmitted to the specimen. The linear energy, measured in J/mm , is:

$$E_l = \frac{U \cdot I}{v} \quad (2.5)$$

Considering the parameters of the first PVR campaign as the nominal energy ($340.2 J/mm$), we chose two additional energy levels for the tests: one of them higher than the nominal energy and another one lower. The low energy level is obtained by reducing the current and the voltage, and augmenting the torch speed. The high energy level is obtained by augmenting the current and the voltage. The used parameters, shown in Table 2.4, were selected after testing the weldability in martyr specimens. Two tests were carried out for each linear energy level configuration.

All tests were instrumented with two type-K thermocouple positioned, as in the 2019 campaign, on both sides of the specimen as shown in Figure 2.3, and an infrared camera. Additionally, the low energy tests were also instrumented with stereo-correlation cameras. This configuration was the only one that ensured that the paint would not burn. In order to allow both infrared and stereo-correlation

TABLE 2.4: PVR experiment parameter values for each energy level.

Energy level	Low	Nominal	High
Current	74 A	81 A	85 A
Voltage	8 V	8.4 V	10 V
Travel speed	2.50 mm/s	2.00 mm/s	2.00 mm/s
Shielding gas	Argon	Argon	Argon
Maximal stroke rate	20 mm/min	20 mm/min	20 mm/min
Linear energy	236.8 J/mm	340.2 J/mm	425 J/mm

observations, only half the specimen is painted, as shown in Figure 2.10. This makes it possible to analyse half of the specimen with the infrared camera without the disturbance produced by the different emissivity of the painted surface.

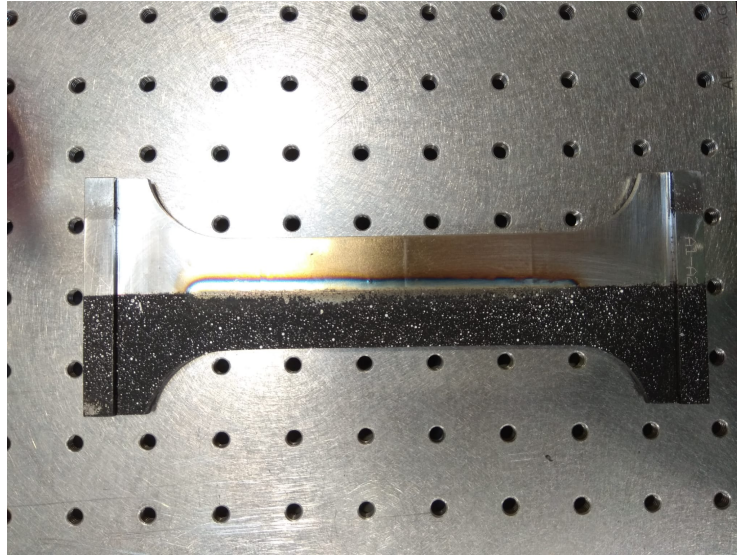


FIGURE 2.10: Half-painted low energy specimen after welding.

The experiments are labelled according to the label that appears on the specimen after fabrication. The specimens are numbered from 1 to 9. Table 2.5 details what each specimen was used for, including the sacrificial specimens. The latter were used to test the low and high energy configurations and verify the proper functioning of the instrumentation.

The duration of each experiment is determined by the speed v and determined by:

$$t_{exp} = \frac{v}{l} + t_{ini} \quad (2.6)$$

where $l = 130 \text{ mm}$ is the length of the weld line and $t_{ini} = 2 \text{ s}$ is the time needed to initiate the torch.

TABLE 2.5: Experiment labels, energy levels and total experiment time.

Specimen number	Energy level	Total time
1	Nominal	67 s
2	Sacrificial	–
3	Sacrificial	–
4	Sacrificial	–
5	Low	54 s
6	High	67 s
7	High	67 s
8	Low	54 s
9	Nominal	67 s

2.3 Digital twin of a PVR experiment

This section describes the creation of a digital twin of the above PVR experiment via Finite Elements Analysis. First, the thermomechanical model is described, detailing the equations, boundary conditions and initial conditions. A second subsection presents the equivalent heat source model and its calibration, which is treated as an optimisation problem between numerical and experimental data.

2.3.1 Thermomechanical model

Thermo-elasto-plasticity

Numerical simulation of welding is a complex problem, as it needs to consider a significant number of parameters to represent multiscale and multiphysics phenomena, using temperature-dependent material properties that are not always precisely known. In the general context of welding simulations, the main phenomena that have to be considered are the strong interactions between the thermal dynamics, metallurgy and mechanics that occur in welding need to be simulated. These interactions are briefly summed up in Figure 2.11, as described in [78]. More details on the nature of these interactions can be found, for example, in [27]. In the case of 316L stainless steel, the metallurgic interactions are negligible [43], and thus, the model is reduced to a weakly coupled non-linear parametric thermo-elasto-viscoplasticity problem.

We consider a model of unsteady thermo-elasto-viscoplasticity over spatial domain $\Omega \in \mathbb{R}^3$, whose boundary will be denoted by $\partial\Omega$, and time domain $\mathcal{T} \in [0, T]$. For all $(x, t) \in \Omega \times \mathcal{T}$, the unsteady heat equation reads as

$$\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot k \nabla T = q_d \quad (2.7)$$

where $T : \Omega \times \mathcal{T} \rightarrow \mathbb{R}$ is the temperature field, ρ is the mass density, k is the thermal diffusion coefficient, c_p is the specific heat capacity, and q_d is the volumetric source term of the heat source.

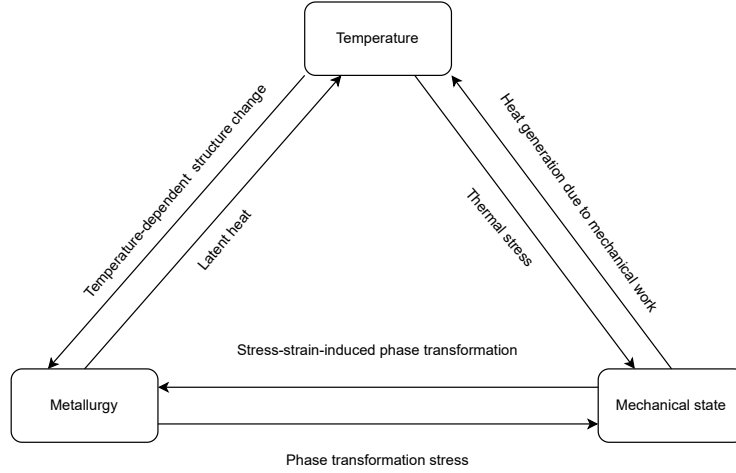


FIGURE 2.11: Interactions between temperature, metallurgy and mechanics in a welding process.

The above equation is complemented by the boundary condition:

$$T = T_0 \quad (2.8)$$

on domain boundary $\partial\Omega_0 \cup \partial\Omega_u$ (the clamping zones, as indicated in Figure 2.1), with $T_0 = 20$ the temperature of the room in which the experiment is carried out. The other boundary condition for the heat problem is the radiation-convection, which is active on the surface of the specimen as:

$$-k\nabla T \cdot n = \gamma(T^4 - T_0^4) + \zeta(T - T_0) \quad (2.9)$$

where γ is the radiation coefficient, and ζ is the thermal convection coefficient.

The entire specimen's temperature is supposed to be equal to room temperature at the beginning of the experiment, so the initial condition is set to:

$$T(x, 0) = T_0 \quad \forall x \in \Omega \quad (2.10)$$

Based on previous work on the simulation of PVR experiments [159], the selected constitutive law used to describe the mechanical behaviour is the Chaboche elasto-visco-plastic model [91, 21]. This law is described by the parameters presented in Table 2.6.

For all $(x, t) \in \Omega \times \mathcal{T}$, the mechanical equilibrium reads as

$$\nabla \cdot \underline{\underline{\sigma}} = 0 \quad (2.11)$$

In the specimen of a PVR test, the traction boundary conditions are enforced by Dirichlet boundary conditions:

$$u(x, t) = u_d(x, t) \frac{t}{t_w} \quad \forall x \in \partial\Omega_u \quad (2.12)$$

on the part of the domain $\partial\Omega_u$, where the tensile loading machine applies the charge. On the other side of the tensile loading machine, the clamping is enforced by:

$$u(x, t) = 0 \quad \forall x \in \partial\Omega_0 \quad (2.13)$$

These boundary conditions are complemented by:

$$\underline{\underline{\sigma}} \cdot \underline{\underline{n}} = 0 \quad (2.14)$$

on the rest of the specimen's surface.

The coupled thermo-mechanical problem is closed by the following constitutive relation:

$$\underline{\underline{\sigma}} = C : (\underline{\underline{\varepsilon}}(u) - \underline{\underline{\varepsilon}}^p - \alpha_{th}(T - T_0)I_d) \quad (2.15)$$

where $\underline{\underline{\varepsilon}}(u) := \frac{1}{2}(\nabla u + \nabla u^T)$ is the total strain, $\underline{\underline{\varepsilon}}^p$ is the plastic strain, and $\alpha_{th}(T - T_0)I_d$ is the thermal strain, with α_{th} the coefficient of thermal expansion and I_d the identity tensor. The plastic strain is fully determined by a Von Mises plasticity model with isotropic and kinematic hardening [91].

$$f(\underline{\underline{\sigma}}, R(p), \underline{\underline{X}}) := \sqrt{\frac{3}{2}(\tilde{\underline{\underline{\sigma}}} - \underline{\underline{X}}) : (\tilde{\underline{\underline{\sigma}}} - \underline{\underline{X}})} - R(p) \leq 0 \quad (2.16)$$

$$\dot{\underline{\underline{\varepsilon}}}^p = \lambda \frac{df}{d\underline{\underline{\sigma}}} \quad (2.17)$$

$$\dot{p} = \lambda \frac{dR}{dp} \quad (2.18)$$

$$R(p) = R_\infty + (R_0 - R_\infty)e^{-bp} \quad (2.19)$$

$$\underline{\underline{X}} = \frac{2}{3}C(p)\underline{\underline{\alpha}} \quad (2.20)$$

$$\dot{\underline{\underline{\alpha}}} = \dot{\underline{\underline{\varepsilon}}}^p - \gamma(p)\underline{\underline{\alpha}}\dot{p} \quad (2.21)$$

$$\gamma(p) = \gamma_0 \quad (2.22)$$

$$C(p) = C_\infty \quad (2.23)$$

$$\lambda \geq 0 \quad \lambda f(\underline{\underline{\sigma}}) = 0 \quad (2.24)$$

where $\underline{\underline{X}}$ is the kinematic hardening tensor and $R(p)$ is the isotropic part of the hardening and is determined by R_0 and R_∞ , which denote the yield stress (expressed in MPa) without cumulative plastic deformation and with infinite cumulative plastic deformation, respectively. λ is the plastic multiplier and $\tilde{\underline{\underline{\sigma}}} = \underline{\underline{\sigma}} - \frac{1}{3}Tr(\underline{\underline{\sigma}})I_d$ is the deviatoric part of the stress tensor.

The main mechanical quantity of interest in the present work is the first principal stress, denoted by σ_I and computed from the stress tensor $\underline{\underline{\sigma}}$ as:

$$\sigma_I = \max_{n \in \mathbb{R}^3} \frac{n \cdot \underline{\underline{\sigma}} \cdot n}{\|n\|^2} \quad (2.25)$$

The first principal stress is the highest principal stress that appears in the same direction as the weld.

TABLE 2.6: Temperature dependent parameters of the elasto-viscoplastic Chaboche model.

T ($^{\circ}C$)	R_0 (MPa)	R_{∞} (MPa)	b	K	N	C_{∞} (MPa)	γ_0
20	60	80	8	151	24	3×10^4	350
600	10	79.4	8	150	12	2×10^4	300
775	9.2	19.8	16.5	179	12.8	8.12×10^3	279
850	2.4	15.8	22.5	298	7.06	1324	154
1000	1	0.2	7.97	192	5.25	244.6	26.3
1150	0.2	0.1	7.97	111	4.98	0	26.3
1250	0.1	0.1	7.97	70	4.91	0	26.3

Thermomechanical load

A volumetric equivalent heat source model is introduced into the heat equation in order to simulate the heat transfer between the torch and the specimen. The choice of a model for the moving heat source is crucial in the numerical simulation of welding. The most commonly used one is Goldak's double ellipsoid model [61, 60], represented in Figure 2.12.

Notice that, as a result of the advancement of the torch, the heat distribution is different in the front and the rear of the heat source; thus, the model depends on the current central position of the heat source $P(t) = (X(t), Y(t), Z(t)) \in \mathbb{R}^3$, which is given by:

$$P(t) = P_0 + \left(\frac{t \cdot v}{(P_F - P_0)} - \epsilon_P(t) \right) (P_F - P_0) \quad (2.26)$$

In the equation above, $P_0 \in \mathbb{R}^3$ is the position of the heat source at the start of the welding operation, and $P_F \in \mathbb{R}^3$ is the position at the end of the operation. $T_{simu} = \frac{(P_F - P_0)}{v}$ corresponds to the total welding time, which depends linearly on the velocity v of the source. $\epsilon_P(t)$ is a slight (nondimensional) time delay that accounts for potential errors in the control of the welding robot and will be estimated online. This model considers the velocity constant during the whole experiment, which is realistic in the case of a robotic arm. However, it can be easily modified for variable velocity.

Assuming that the heat source moves in the direction x , the final equation is

$$q(x, y, z, t) = \begin{cases} \frac{12\sqrt{3}\eta Q}{(a_r + a_f)bc\sqrt{\pi^3}} \exp\left(\frac{-3(x-X)^2}{a_f^2} + \frac{-3(y-Y)^2}{b^2} + \frac{-3(z-Z)^2}{c^2}\right), & x \geq X. \\ \frac{12\sqrt{3}\eta Q}{(a_r + a_f)bc\sqrt{\pi^3}} \exp\left(\frac{-3(x-X)^2}{a_r^2} + \frac{-3(y-Y)^2}{b^2} + \frac{-3(z-Z)^2}{c^2}\right), & x < X. \end{cases} \quad (2.27)$$

where a_r , a_f , b and c are unknown parameters describing the geometry of the double ellipsoid, $Q = UI$ is the power, with U the voltage, I the current and η the efficiency. X , Y and Z correspond in this equation to the current position of the heat source, $P(t)$. The equation depends on the position of a point $(x, y, z) \in \Omega$ at time t with respect to the current torch position, given by $P(t) = (X, Y, Z)$.

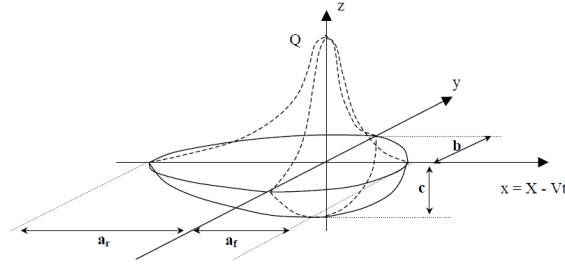


FIGURE 2.12: Goldak's double ellipsoid model.

Space and time discretization

The mesh used for the simulations is a quadratic mesh and it can be observed on Figure 2.13. It contains 104101 nodes. The different levels of mesh refinement in the centre of the specimen to adequately capture the heat evolution around the weld line. The refinement can be observed in Figure 2.13. The specimen is symmetrical, which means that only half of the specimen could have been modelled. Instead, we decided to represent the entire specimen so that more experimental data from the infrared camera could be projected onto the mesh.

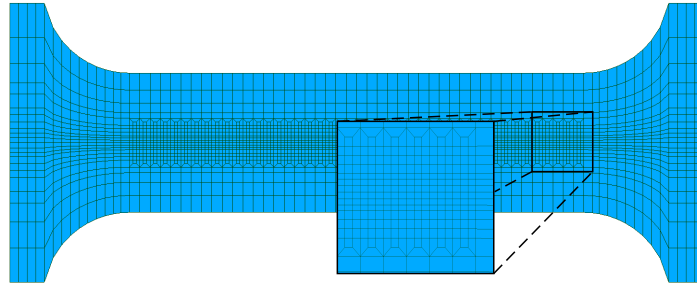


FIGURE 2.13: PVR mesh.

The resolution of the thermo-mechanical problem presented in the previous section is done using the Finite Elements Method. A detailed description of the method and the steps needed to obtain the weak formulation of the problem from the physical equations can be found in general Finite Elements books such as [48, 14]. For details on the implementation in code_aster, EDF's open-source thermo-mechanical Finite Elements software, we refer to its documentation [1].

2.3.2 Design of experiment

The previous sections presented the parameters of the PVR experiment and the numerical model. The PVR experiment can be determined by the operational parameters related to the welding torch: speed (v) and power (Q). Let us remind that the maximum displacement rate could also be an operational parameter, but it is fixed for all the experiments, so it is discarded. Goldak's equivalent heat source model introduces five more parameters relative to the geometry of the double ellipsoid and the efficiency: a_f , a_r , b , c , and η . In practice, we will calibrate the following

parameters: a_f , b , c , η , and K , where $K = \frac{a_r}{a_f}$ is the ratio between the front and rear lengths of the ellipsoid.

For the data assimilation applications presented in future chapters, these parameters are divided into two separate groups: the known parameters $\mu = (v, Q) \in \mathcal{M} \subset \mathbb{R}^2$ and the unknown parameters $\theta = (a_f, b, c, \eta, K) \in \Theta \subset \mathbb{R}^5$. Thus, the parameter space \mathcal{P} can be defined as the product of two distinct blocs \mathcal{M} and Θ :

$$\mathcal{P} = \mathcal{M} \times \Theta \subset \mathbb{R}^7 \quad (2.28)$$

The parameter values are bounded by welding experts. Determining the possible values for the operational parameters μ is no easy task. The experts need to find a combination that ensures weldability but does not pierce the specimen, often by testing the parameters on sacrificial specimens. Numerically, the operational parameter bounds are depicted in Table 2.7.

TABLE 2.7: Minimum and maximum values for the PVR operational known parameters μ .

	Minimum value	Maximum value
Current	70 <i>A</i>	90 <i>A</i>
Voltage	8 <i>V</i>	12 <i>V</i>
Power	560 <i>W</i>	1080 <i>W</i>
Speed	1 <i>mm/s</i>	3 <i>mm/s</i>

The unknown parameters θ are also bounded by experts. The selected values are shown in Figure 2.8. This allows to define more precisely the parameter space \mathcal{P} . As a result, \mathcal{M} and Θ now read:

$$\mathcal{M} = [1.0, 3.0] \times [560, 1080] \subset \mathbb{R}^2 \quad (2.29)$$

$$\Theta = [3, 12] \times [1, 7] \times [0.5, 3] \times [0.75, 0.95] \times [1.15, 2.5] \subset \mathbb{R}^5 \quad (2.30)$$

TABLE 2.8: Minimum and maximum values for the unknown parameters θ , as determined by the experts.

	a_f	b	c	η	K
Min. value	3 <i>mm</i>	1 <i>mm</i>	0.5 <i>mm</i>	0.75	1.15
Max. value	12 <i>mm</i>	7 <i>mm</i>	3 <i>mm</i>	0.95	2.5

Considering the boundaries of each parameter value given in Table 2.7 and Table 2.8, all parameters are supposed to follow a uniform distribution between their extreme values:

$$v \sim \mathcal{U}(1, 3) \quad (2.31a)$$

$$Q \sim \mathcal{U}(560, 1080) \quad (2.31b)$$

$$a_f \sim \mathcal{U}(3, 12) \quad (2.31c)$$

$$b \sim \mathcal{U}(1, 7) \quad (2.31d)$$

$$c \sim \mathcal{U}(0.5, 3) \quad (2.31e)$$

$$\eta \sim \mathcal{U}(0.75, 0.95) \quad (2.31f)$$

$$K \sim \mathcal{U}(1.15, 2.5) \quad (2.31g)$$

The parameter space \mathcal{P} can be explored using one of many Design of Experiments (DOE) techniques such as Latin Hypercube Sampling [52] (LHS) or low discrepancy methods [87]. The simulations cannot be calibrated individually because of the lack of experiments for every single combination of parameters. The DOE has to be performed on a joint uniform distribution calculated from Equations (2.31).

2.3.3 Calibration of Goldak's model

The unknown parameter values of Goldak's equivalent heat source model need to be calibrated before simulating a welding experiment. Mistakes in the values of these parameters lead to vastly different heat distributions in the specimen (see Figure 2.14 in the following section for an example of 400 temperature trajectories computed with 400 different parameter values). The mechanical behaviour depends strongly on the current temperature of the metal, and it cannot be correctly estimated without proper modelling of the heat source.

In some research work, the equivalent heat source's parameters are estimated by measuring the weld pool, as recommended by Goldak [60]. EDF's usual practice is to calibrate the equivalent heat source involved in the FE models with thermocouple experimental data. Calibrating the simulations is meant to ensure a better correspondence between experimental and simulation data. The calibration is presented as a least-squares optimisation problem between the thermocouple data and temperature data extracted from simulations. In particular, we have heeded the following procedure to calibrate the simulation of PVR experiments realised at the laboratory.

In order to calibrate the parameters, we will minimise the discrepancy between the experimental thermocouple measurements and the output of a Finite Elements model which depends on θ and is evaluated at the same position as the thermocouple. The thermocouple data is considered the true value of the temperature and is represented by the vector T_{th} . The dimension of T_{th} depends on the amount of time the thermocouple has been active and the acquisition frequency. To simplify this issue, the thermocouple data is systematically resampled to represent the same N_t time steps as the simulations. Thus, $T_{th} \in \mathbb{R}^{N_t}$ and the minimisation problem is between two vectors of the same dimension:

$$\theta^* = \arg \min_{\theta \in \Theta} \|T_{th} - T(\theta)\|_2 \quad (2.32)$$

where $\|\cdot\|_2$ is the euclidean distance, Θ is the parameter space defined in Equation (2.30) and $T(\theta) \in \mathbb{R}^{N_t}$ is a vector containing the output of the Finite Elements model at the desired position.

The resolution of this problem involves repeated calls to the Finite Elements model, which would be highly costly. Using surrogate models is a way to avoid this particular problem. Here, the output of the Finite Elements simulation is approximated by a Polynomial Chaos Expansion (PCE) [26, 153]. PCE is a regression method based on spectral decomposition of L^2 functions. Let us assume that one such function, h , is the output of interest which depends random variables $X = (X_1, X_2, \dots, X_N)$. PCE uses a basis of orthonormal polynomials $\{\psi_j(X)\}_{j \in \mathbb{N}}$ on a Hilbert space to represent the output of h . The PCE is written:

$$h(X) = \sum_{j \in \mathbb{N}} y_j \psi_j(X) \quad (2.33)$$

which is truncated to a finite set $J \subset \mathbb{N}$:

$$h(X) = \sum_{j \in J} y_j \psi_j(X) \quad (2.34)$$

where y_j are coefficients that characterize the output of $h(X)$ and which need to be estimated using simulation data.

In our case, the output corresponds to the temperature trajectories resulting from a FE simulation $T(\theta)$, which depend on θ . A function f will describe the behaviour of the trajectories as a series of polynomials. Following the notation in Equation (2.34), the surrogate model is:

$$T(\theta) \approx f(\theta) = \sum_{j \in J} y_j \psi_j(\theta) \quad (2.35)$$

The PCE needs the simulation parameters θ as input. In order to cover the entire θ -space, the first step is to define a design of experiment. Considering the boundaries of each parameter value given in Table 2.8, all parameters are supposed to follow a uniform distribution between their extreme values, as presented in Equation (2.31). Using LHS, $M = 400$ vectors $\theta_1, \theta_2, \dots, \theta_M$ are sampled. A whole heat simulation is computed for each sample setting the operational parameters as $v = 2.0 \text{ mm/s}$ and $Q = 850 \text{ W}$, which corresponds to the high energy configuration of the experiment. The temperature values at the thermocouple position are extracted for all time steps, yielding a total of M temperature trajectories $T(\theta_1), T(\theta_2), \dots, T(\theta_M)$ where each $T(\theta_i)$ is a vector of dimension N_t . More details on how the simulations are carried out are given in the next section.

The resulting surrogate model might have a large dimension if the simulation is computed for many time steps. To simplify the problem, the dimensionality of the data can be reduced by a standard Singular Value Decomposition (SVD) performed on a matrix containing temperature trajectories computed via Finite Elements simulation for different θ values. The trajectories are stored in a matrix \mathcal{S} containing the M trajectories:

$$\mathcal{S} = [T(\theta_1), T(\theta_2), \dots, T(\theta_M)] \in \mathbb{R}^{N_t \times M} \quad (2.36)$$

where each line is one of the trajectories. The SVD reads

$$\mathcal{S} = V\Lambda W^T \quad (2.37)$$

Selecting the reduced dimension $l < N_t$ is always a key when applying dimensionality reduction. Here it is chosen to represent at least 95% of the variance. The variance ratio, or sometimes called energy ratio [166], measures how much of the variance (or energy) is explained by the selected modes with respect to the total variance. 95% of variance corresponds to a ratio of $\mathcal{E}(l) = 0.95$, where $\mathcal{E}(l)$ is:

$$\mathcal{E}(l) = \frac{\sum_{i=1}^l \lambda_i}{\sum_{i=1}^{N_t} \lambda_i} \quad (2.38)$$

where λ_i are the singular values ordered from largest to smallest. The temperature trajectories are projected into the p -dimensional space in order to obtain their reduced representation:

$$\tilde{T}(\theta) = V_p^T T(\theta) \quad (2.39)$$

where V_p is a matrix composed of the first p singular vectors. In order to quantify the performance of the dimensionality reduction, the relative reconstruction error of \mathcal{S} , the matrix containing the M trajectories, can be computed as:

$$e_r = \frac{\|\mathcal{S} - V_p(V_p^T \mathcal{S})\|_2}{\|\mathcal{S}\|_2} \quad (2.40)$$

where $\|\cdot\|_2$ denotes the euclidean norm.

After the dimensionality reduction, the output of the surrogate model is a p -dimensional vector. The PCE will now be a function $\tilde{f} : \Theta \subset \mathbb{R}^5 \rightarrow \mathbb{R}^p$ that approximates the reduced temperature trajectories :

$$\tilde{T}(\theta) \approx \tilde{f}(\theta) = \sum_{j \in J} y_j \psi_j(\theta) \quad (2.41)$$

where p is the first reduced dimension such that $\mathcal{E}(p) \geq 0.95$. The predictivity factor Q_2 of the PCE can be used as an estimator of the performance of the surrogate model. The Q_2 score is the classic R_2 score applied to the prediction residuals [107]. It is given by:

$$Q_2 = 1 - \frac{\sum_{i=1}^N \left(\tilde{T}(\theta_i) - \tilde{f}(\theta_i) \right)^2}{N \cdot \text{Var}(\tilde{T}(\theta))} \quad (2.42)$$

The optimisation problem is reduced to the minimisation of the distance between two vectors of dimension p . Indeed, the experimental data can also be projected onto the reduced dimension space. The p -dimensional representation of the experimental data is written T_{th}^p . The final optimisation problem is:

$$\theta^* = \arg \min_{\theta \in \Theta} \|T_{th}^p - \tilde{f}(\theta)\|_2 \quad (2.43)$$

A standard BFGS algorithm [116] is used to solve the problem.

Example of calibration

This section shows the calibration process on one of the experiments of the 2021 campaign, in particular specimen number 6. According to Table 2.5, Test number 6 is a high energy test.

Latin Hypercube Sampling is used to sample $M = 400$ θ vectors $\theta_1, \theta_2, \dots, \theta_{400}$. The 400 hundred temperature trajectories $T(\theta_1), T(\theta_2), \dots, T(\theta_{400})$ are obtained from a simulation with $N_t = 260$ times steps. Thus, each $T(\theta_i)$ is a vector of dimension 260. We remind that the position of the thermocouple in this case is 110 mm from the bottom and 6 mm to the right of the specimen's centre (see Figure 2.3). These 400 trajectories are shown in Figure 2.14, where it is clear that θ has a significant influence on the temperature profile in the simulation.

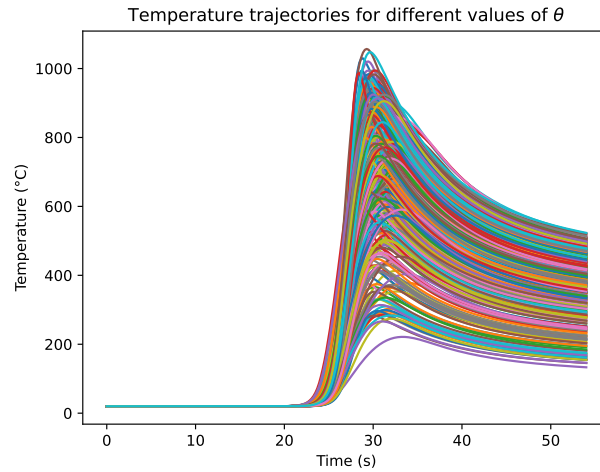


FIGURE 2.14: Temperature values at the thermocouple position for 400 samples of θ .

With the SVD, the dimension of the data can be reduced to $p = 2$, which corresponds to 96% of the variance, as seen in Fig 2.15. The reconstruction error, presented in Equation (2.40), of \mathcal{S} is 2.24% whilst the reconstruction error of the thermocouple's data is 3.61%.

The surrogate model links a vector θ with the 2-dimensional representation of the temperature trajectories. To construct the PCE, the data is randomly divided into a training set with 80% of the data and a testing set with 20%. The surrogate model is computed using the Open TURNS library [20], which is developed at EDF R&D. According to Equation (2.31), the distribution of each element of θ is uniform. Thus the polynomials used for the PCE are Legendre polynomials [26].

The final Q_2 prediction score of the surrogate model is excellent, as it is extremely close to 1:

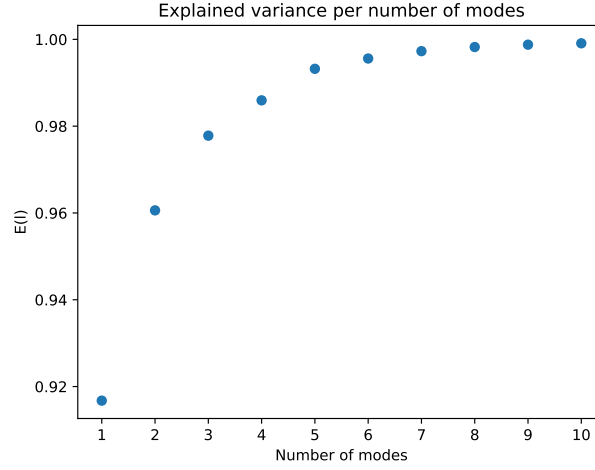
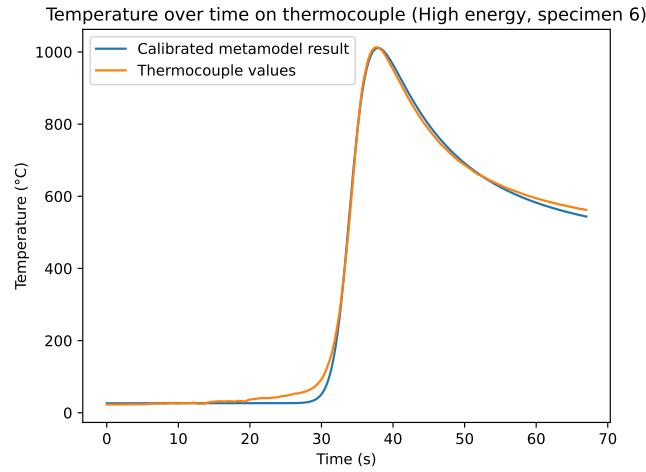


FIGURE 2.15: Explained variance per number of modes in SVD.

$$Q_2 = 0.99995 \quad (2.44)$$

The surrogate model can be used for further Uncertainty Quantification (UQ). For example, to compute Sobol Indices [145] to determine the most influential parameters. These additional studies are out of scope for this chapter and thus will not be treated. The surrogate model is then used for the resolution of the minimisation problem formulated in Equation (2.43). We have used a BFGS algorithm [116] with multiple start points in order to converge towards the solution. The convergence of the optimisation problem allowed to identify the following Goldak's parameters:

$$\theta = (5.69, 5.24, 2.71, 0.95, 1.99) \quad (2.45)$$

FIGURE 2.16: Thermocouple and surrogate model result for the optimal θ .

These parameters have been used to simulate the high energy experiment of the 2021 campaign. The surrogate model applied on the optimal θ is shown in Figure 2.16, which is very close to the thermocouple time series. Indeed, the relative L^2 error between these curves is 2.94% and the relative L^∞ error is 4.17%.

Calibration methodology

The following list summarises the main steps of the heat source calibration:

Calibration methodology

- 1 Design of Experiment for an adequate exploration of the parameter space.
- 2 Simulate a large number of cases and extract the temperature trajectories at the thermocouple position.
- 3 Compute an SVD to reduce the dimension of the problem.
- 4 Create a surrogate model that mimics the temperature trajectories produced by simulation.
- 5 Minimise the distance between the thermocouple and the output of the surrogate model to find the optimal parameters.

2.4 Parametric study of a PVR experiment

This section describes the generation of a numerical database from a parametric study of a PVR experiment.

2.4.1 Design of Experiment

The objective of the parametric study is to represent a large variety of parametric solutions to the thermo-mechanical problem in order to create reduced local bases capable of representing them. A total of 128 samples are obtained by using LHS on a joint uniform distribution that includes the random variables described in Equations (2.31).

2.4.2 Simulation

The simulations are run on a scientific cluster with code_aster, EDF's open-source thermomechanical FE software [47]. As said previously, numerical simulation of welding with 316L steel can be treated as a weakly coupled heat and mechanical problem where the heat problem is solved first, and the result is used as input for the mechanical problem.

The simulations run for $N_t = 260$ time steps that vary with the speed. In all cases the welding line is 130 mm long, which means that for each time increase Δt the torch has advanced 0.5 mm. The time grid is:

$$\bar{\mathcal{T}} = \{t^0 = 0, t^1 = \Delta t, t^2 = 2\Delta t, \dots t^{N_t} = N_t \Delta t\} \quad (2.46)$$

where Δt depends on the speed v :

$$\Delta t = \frac{(P_F - P_0)}{N_t v} \quad (2.47)$$

The final results are, thus, computed with a different Δt for each speed value, but the stored data always refers to the same positions of the torch's advance, which will be of crucial importance in the following chapter.

Table 2.9 presents the average computation time of the thermo-mechanical simulations, separated into the thermal and mechanical steps of the weakly coupled problem:

TABLE 2.9: Number of CPUs and average computation time of the simulations.

	Number of CPUs	Average computation time
Heat problem	8	1 h 50 min
Mechanical problem	48	7 h 56 min

2.4.3 Results

The parametric study results in a large simulation database of temperature and mechanical fields that will be used in the following chapters. Figures 2.17 and 2.18 present examples of the calibrated simulations of the 2019 experiment. The temperature simulation (2.17) shows the progression of the welding torch, located in the centre of the specimen. Figure 2.19 displays the predicted molten zone on the PVR specimen's cross-section, corresponding to the same snapshot.

A usual analysis of PVR experiments is the comparison of the force/displacement profiles [160]. Using the experimental data obtained in the 2021 campaign, Figure 2.20 shows the comparison between the simulations and the machine results. Despite using the same model (and software) presented by Tran Van in [160], the experimental and simulated profiles do not coincide. The only difference between Tran Van's case and the one presented in this PhD is the size of the specimen, which has been shortened from 260 *mm* to 200 *mm*. The discrepancy between the experimental and simulated data was identified after finishing the 2021 campaign. The main reason that explains why this constitutive law might not work as intended might be the existing fitted implementation of the law in `code_aster`. Fitting the Chaboche constitutive behaviour is a difficult task that needs monotonic or cyclic thermo-mechanical experimental data [152], and the fitted implementation in `code_aster` might not be well-suited for PVR experiment's simulations. Not being the main objective of the PhD and after having spent some time searching for possible solutions without success, it was decided to keep using this model and show the results as they are. This is an opportunity to point out the dependence between the ability of the FE models to correctly represent the problem's physics and the accuracy of

the results for the data assimilation approaches presented in the following chapters. Especially in the current case, where no observations of the mechanical fields are made.

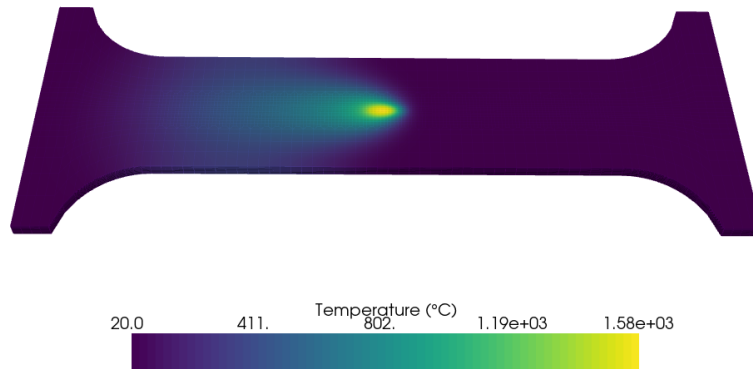


FIGURE 2.17: Snapshot of a temperature calibrated simulation of the 2019 experiment.

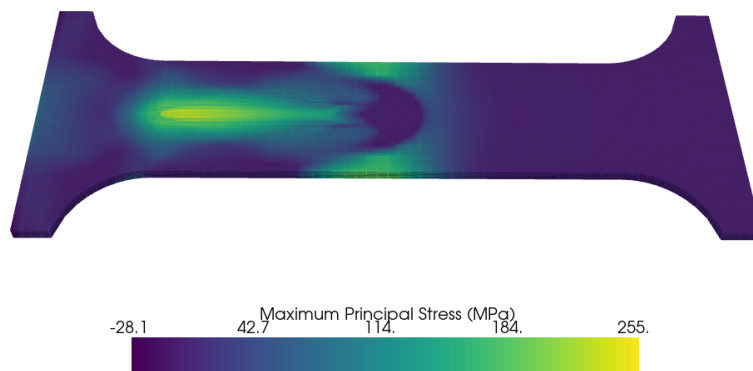


FIGURE 2.18: Snapshot of a principal stress field of a calibrated simulation of the 2019 experiment.

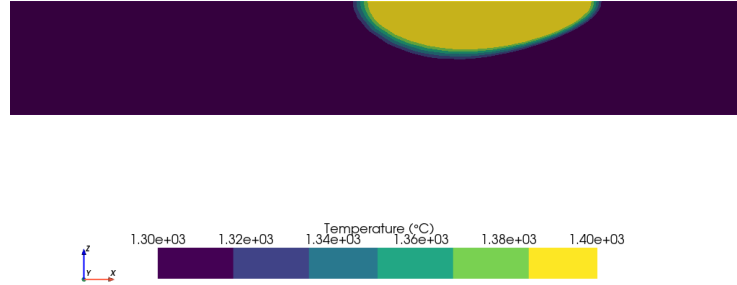


FIGURE 2.19: Cross-section view of the molten zone of a temperature snapshot of the 2019 experiment.

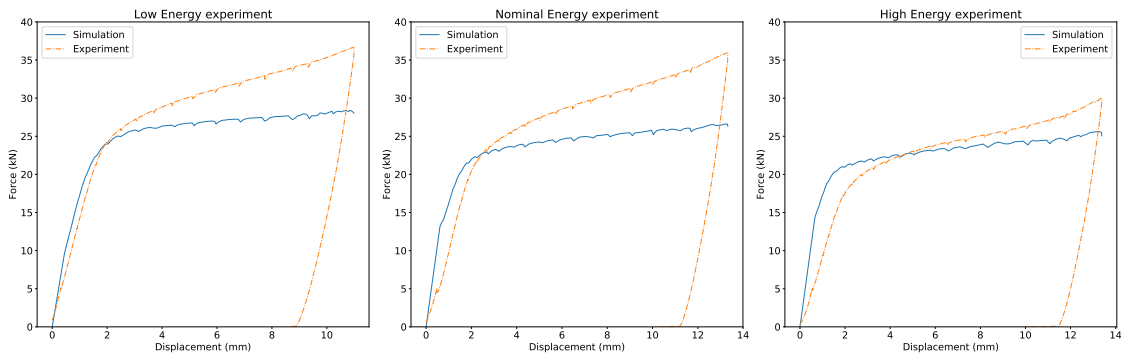


FIGURE 2.20: Force/displacement simulation and experimental profiles of the 3 energy levels.

2.4.4 Dimension and size of the generated data

Let us underline the dimensions of the involved DOE and the size of the saved files. The DOE contains 128 simulations. For each, the computational domain contains 104101 spatial nodes and 260 time steps. The simulation output is made of two result files, one for the heat problem and another one for the mechanical problem. Each heat solution file is 844 *Mb* and each mechanical solution file is 33 *Gb*. The whole parametrical study occupies more than 4 *Tb*.

Not all the data generated during the simulations are needed for the inverse problems applications explained in future chapters. For an efficient treatment of the data, only the desired fields are stored and compressed in HDF5 [157] containers, that allow for high-speed access and reduced storage space. This way, only saving temperature and nodal stress (the projection from the Gauss integration points to

the nodes), the heat solution files only occupy, on average, 126 *Mb* and the mechanical results occupy, on average, 2.9 *Gb*.

2.5 Merging experimental and simulation data

This chapter has explained the experiments that were carried out during the PhD and how to create their high-fidelity digital twins. All the generated data will be used as input for a data assimilation problem or to generate models in future chapters. However, the interaction between simulation and experimental data requires some additional treatment. This last section shows two particular instances of this treatment: estimating the infrared camera's measurement error and the projection of experimental data to the Finite Elements mesh.

2.5.1 Estimating the infrared camera measurement error

Determining the measurement error of the camera is of the most importance for the data assimilation problem that will be presented in the following chapter. The actual temperature cannot be measured without noise, neither with the thermocouples nor the camera. The camera manufacturer gives a measurement error estimated on a black body calibrator. However, without accurate knowledge of the specimen material's emissivity, this error is not applicable to our situation. Our proposal is to estimate the measurement error of the camera as the discrepancy between both sources of experimental data.

The camera measures temperature on a surface whilst the thermocouple offer a punctual measurement. The first step consists of identifying the thermocouple measurement point on the image and extracting the temperature over time. Figure 2.21 shows both measurements obtained from the 2019 experiment. One can observe that the measurements are quite similar for both instruments for temperatures above 300°C. The exception is a slight peak at the beginning of the cooling phase that can be attributed to a reflection that creates a visual artefact, which is interpreted as a higher temperature.

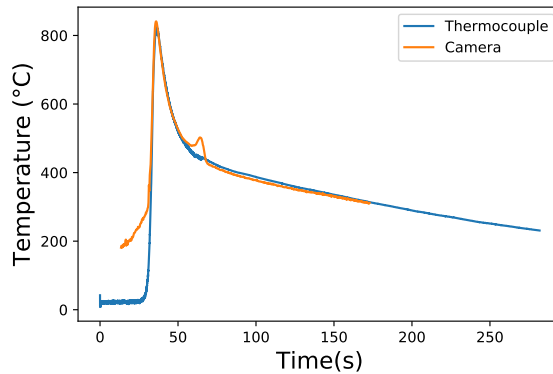


FIGURE 2.21: Temperature measures taken with a thermocouple and an infrared camera obtained from the 2019 experiment.

Both curves are resampled to take into account the different acquisition times. All temperatures below 400°C are discarded for the analysis because the camera is not calibrated to measure in that temperature range correctly. The absolute error between both measurements is stored in a variable η :

$$\eta = T_{cam} - T_{thermo} \quad (2.48)$$

where T_{cam} and T_{thermo} are the temperature values measured with the camera and thermocouple, respectively. The time series η is plotted in Figure 2.22.

The measurement error is estimated as the variance of η , which corresponds to a standard deviation of about 18°C for the 2019 experiment. A more robust estimation of the error between the camera and the thermocouples requires more experimental data. However, when the measurement error was calculated, only the 2019 experiment was available.

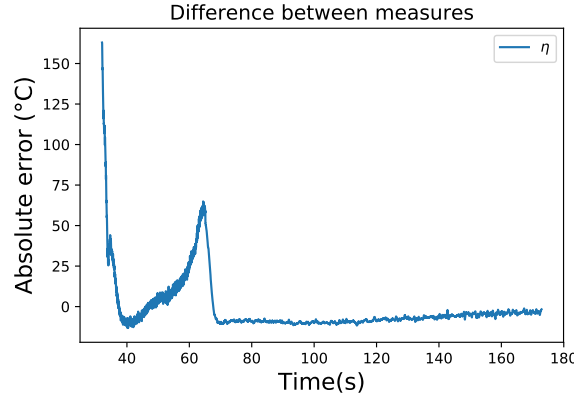


FIGURE 2.22: Difference between camera and thermocouple data over time.

2.5.2 Projection of the infrared camera data into the FE mesh

The methods presented in the following chapters rely on using the infrared images in conjunction with the Finite Elements solutions obtained in the parametric study. The raw images from the camera are expressed in Digital Level signals, a value between 0 and 16000 that represents the luminance going through the filters. The digital level can be transformed into temperature using the software provided by the manufacturer.

Instead of using this software, two of the paired filters are selected, and the raw data is converted into temperature using Planck's law, as seen in Equation (2.4). This allows for a faster and automatised analysis of the images in the same Python environment, which should be helpful for on-the-fly analysis of infrared images and, thus, the estimation of temperature and stress fields. The code that accesses the raw data and transforms it to temperature was developed in collaboration with Théo Boutin for his PhD on the detection of abnormal behaviour in welding [28], and tested on the 2019 PVR experiment and a test case for Sami Hilal's PhD [74]. The overall approach and results are detailed in an internal report [29].

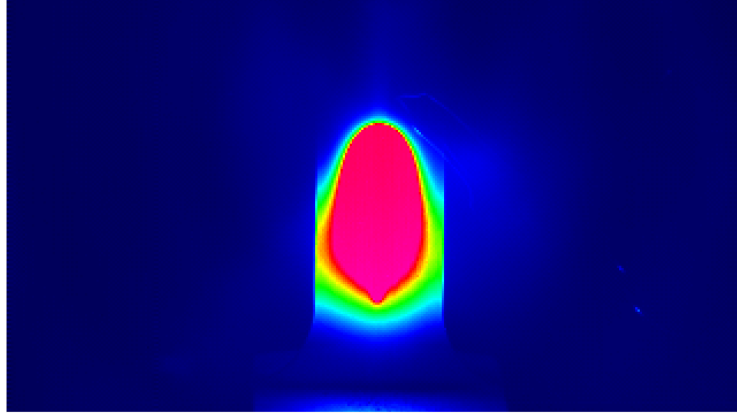


FIGURE 2.23: Raw infrared video frame. Colours represent Digital Levels.

Once the data has been transformed into temperatures, it has to be projected onto the Finite Elements mesh. We will use an intermediate mesh with the same dimensions as the image. This requires the translation of distances in pixels d_p into distances in millimetres d_m . The relationship between pixel and millimetre distances is calculated using an easily identifiable and known feature of the specimen, for example, its width. The width of the specimen is 40 *mm*, and it is easily separated from the background, as seen in Figure 2.23. Thus, the ratio r between distance in pixels and millimetres reads:

$$r = \frac{d_m}{d_p} \quad (2.49)$$

The intermediate virtual mesh is a regular grid created with the same dimensions as the image. This allows to recreate the image in the same referential as the FE mesh, which can be seen in Figure 2.24. As a consequence, the intermediate and PVR meshes lie in the same space, therefore they can be aligned by using a reference point. In this approach, the starting point of the weld is usually selected as the reference point because it is easily identifiable on the infrared images, being the point that lights up when the torch is turned on. Figure 2.25 shows the intermediate mesh on top of the PVR mesh, which is represented by its wireframes.

The temperature field is assigned to the PVR mesh by evaluating the temperature field on the coordinates of the PVR mesh's nodes. This results on a temperature field assigned exclusively on the observed area, shown in Figure 2.26.

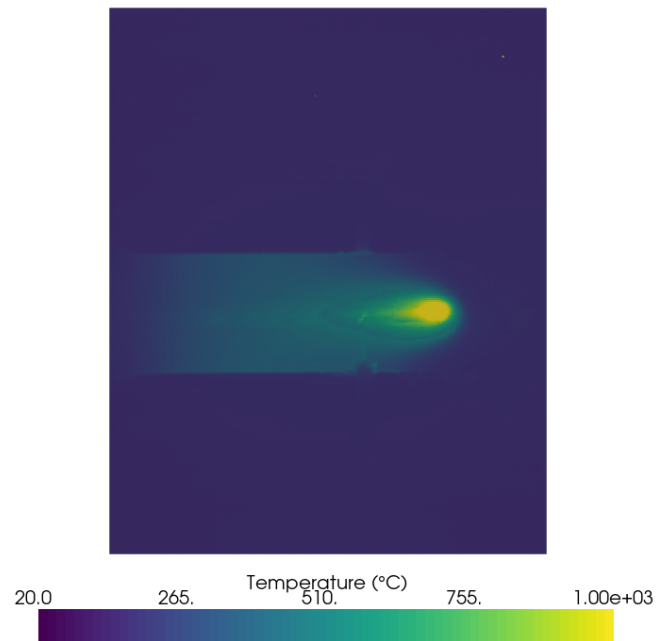


FIGURE 2.24: Temperature values on the intermediate mesh. Example obtained from specimen 9 of the 2021 campaign.

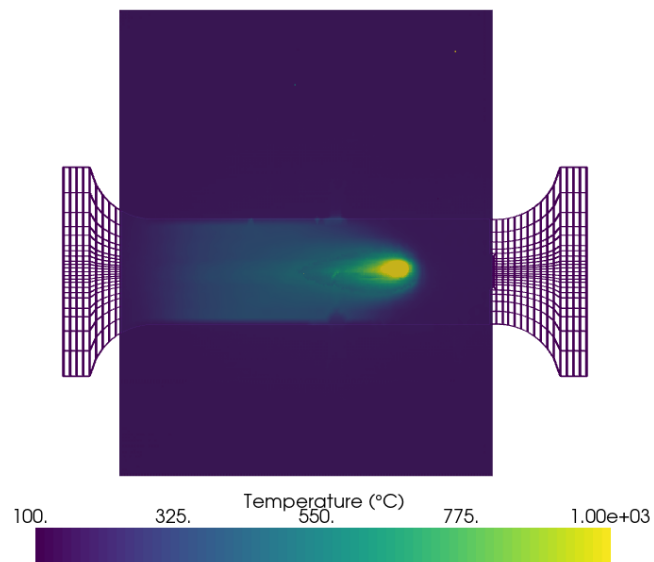


FIGURE 2.25: Superposition of the intermediate and the PVR meshes. Example obtained from specimen 9 of the 2021 campaign.

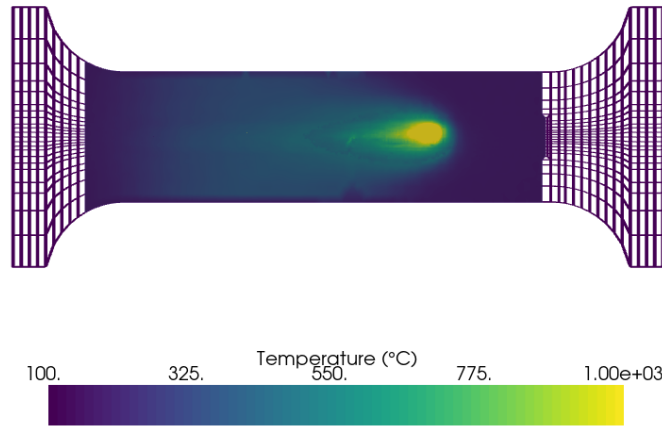


FIGURE 2.26: Infrared temperature values on the PVR mesh. Example obtained from specimen 9 of the 2021 campaign.

2.6 Conclusion

This chapter has presented the PVR experiments realised at EDF R&D and how to create their digital twins. Although the objective of the PhD is not to study hot cracking issues, the PVR experiments are a practical use case for data assimilation problems in welding. Indeed, a great variety of instrumentation can be deployed during the experiments in order to obtain experimental data. Such data have been obtained from three different sources: thermocouples, infrared cameras and stereo-correlation cameras. The main advantage of the PVR experiment with respect to industrial welding operations is its ability to control parameters while using the same material and process.

Two PVR experimental campaigns were carried out at EDF R&D welding lab. The different operational parameters were introduced and discussed. The experimental configuration and the instrumentation of each experiment were also presented.

Simulation-based digital twinning depends on many external factors that are not well controlled, such as material properties and modelisation. The calibration of the equivalent heat source model is a critical step in the numerical simulation of welding for an optimal correspondence between the results and the experimental data. The calibration procedure involves the resolution of an optimisation problem. A batch of 400 simulations is computed in order to generate a Polynomial Chaos surrogate model that mimics temperature trajectories. Dimensionality reduction is used to

simplify the construction of the surrogate models. The surrogate model is then used to generate a larger number of temperature trajectories.

The parametrical study of the PVR experiment is the basis for the offline-online approach presented in the next chapter. With a very refined mesh as the one used for these studies, the dimension of the temperature and mechanical fields is very large. Dealing with large amounts of data is a challenge in itself, especially when the goal is to obtain results in real time.

The data assimilation problem that will be tackled in the following chapters requires the interaction between the simulation and experimental data, in particular, the infrared images. We have shown a custom-made method that allows to extract temperature values from luminance levels on the infrared image and the projection of the infrared images onto the FE mesh. The measurement error of the camera needs to be estimated. Indeed, the experimental configuration of the PVR experiment is such that the measurement error given by the manufacturer is not applicable. An estimation of this error as the variance of the difference between two time series has been presented in this chapter.

Chapter 3

Local Probabilistic PCA surrogate models

Abstract:

This chapter will present our offline-online approach to solving a welding data assimilation problem where not all the desired objective quantities are observable. Here, most observable data are obtained by an infrared camera. First, we will introduce the “truth” inverse problem that we aim to solve and present the infeasibility of its resolution in real time. The following section will explain the offline stage of the approach: detailing the construction of the local multiphysics Probabilistic PCA model and how to obtain them from a Finite Elements parametric study. Then, we will show how to solve the new reduced data assimilation problems in real-time using Bayesian inversion and focusing on the hurdles we need to overcome: the fast computation of the posterior distribution and the identification of the adequate model. A brief section will introduce the notion of forecasting with these local models. Finally, we will show some numerical results on the PVR experimental case.

The contents of this chapter were adapted from an article published in 2021 [[122](#)].

Résumé :

Ce chapitre présente une approche offline-online pour résoudre un problème d’assimilation de données en soudage où une partie des quantités d’intérêt ne sont pas observables. Ici, la plupart des données observables sont obtenues à l’aide d’une caméra infrarouge. Premièrement, on introduira le “vrai” problème inverse que l’on essaie de résoudre et on montrera pourquoi sa résolution en temps réel n’est pas faisable. La section suivante traitera la partie offline de notre approche en détaillant la construction d’un modèle local d’Analyse en Composantes Principales Probabiliste (ACPP) multiphysique à partir d’une étude paramétrique par Éléments Finis. Puis, on montrera comment résoudre les problèmes réduits d’assimilation de données en temps réel par inversion Bayésienne, en se concentrant sur les principaux verrous techniques : le calcul rapide de la distribution postérieure et l’identification du bon modèle à utiliser. Une brève section introduira, en suite, la notion de la prévision d’instant

futurs et comment le réaliser avec ces modèles. Finalement, on montrera des résultats de tests numériques sur le cas test expérimental PVR.

Le contenu de ce chapitre a été adapté à partir d'un article publié en 2021 [[122](#)].

Contents

3.1 Truth inverse problem	62
3.1.1 Parametrised probabilistic setting	62
3.1.2 "Truth" online Bayesian conditioning	63
3.2 Local multiphysics PCA model	64
3.2.1 Local surrogate models	65
3.2.2 Maximum likelihood PCA	67
3.2.3 Offline stage methodology	68
3.3 Online prediction in real time	68
3.3.1 Fast computation of mean and covariance	69
3.3.2 Known position	71
3.3.3 Unknown position	72
3.3.4 Online stage methodology	73
3.4 Forecasting of future states	74
3.4.1 Forecasting methodology	74
3.5 Numerical results	75
3.5.1 Estimation of the heat source position	75
3.5.2 Tests with noisy synthetic data	77
3.5.3 Tests with real experimental data	79
3.6 Conclusion	85

3.1 Truth inverse problem

As we saw in Chapter 1, real-time quality control of welding operations would be a game-changer for the nuclear industry. Some welding defects can be identified using temperature and mechanical criteria [160]. In most cases, the mechanical fields are not observable during the welding operation. Other physical quantities might be observable but do not give enough information to describe the state of the welding. One such quantity is temperature, which can be measured with thermocouples and infrared cameras.

Numerical simulation of welding is a powerful tool that could give access to the desired fields, such as stress and strain, but the complexity of the thermomechanical models makes digital twinning unfit for real-time estimation. The use of Model Order Reduction (MOR) techniques such as the hyper-reduction method [140, 141, 45, 176, 37] would be helpful, but it does not solve the problem of identifying the problem parameters on the fly from an observable physical quantity. This section will introduce the parametrised probabilistic setting of the problem. Then, the inverse problem will be formulated, and its resolution will be discussed.

3.1.1 Parametrised probabilistic setting

In Chapter 2, the simulation and experimental data were presented. We saw that the thermomechanical Finite Element models are heavily parametrised, including thermal and mechanical temperature-dependent parameters, operational parameters, boundary conditions, and the volumetric equivalent heat source model. Correctly identifying these parameters is necessary because they introduce uncertainty into the model.

The three main operational parameters, the heat source's velocity (v), voltage U , and current (I), are supposed to be controlled with a reasonable degree of accuracy. Those are the input settings of the robotic arm and are called, from now on, the known parameters μ . Usually, U and I are presented together as the heat source power $Q = UI$. The mechanical load u_d is also well controlled during the experimental procedure. However, it will not be considered as one of the known parameters because none of the experiments realised at the welding lab changed the tensile load u_d and is, thus, fixed in all the simulations. In order to build the "truth" digital twin, we further assume that the thermal capacity and diffusivity of the material are well characterised. The mechanical behaviour of the structure (thermal expansion, elasticity and plasticity) are also assumed to be well characterised, qualitatively and quantitatively. These assumptions are consistent with EDF's decades of experience in modelling, characterising and simulating such welding processes. Finally, the thermal and mechanical boundary conditions are reasonably well quantified in our experimental setting.

However, several sources of uncertainties negatively affect the predictive capabilities of the simulation model. These sources are listed below:

- the position P of the centre of the heat source is not perfectly well known at all times. This lack of knowledge is represented by a random variable ϵ_P of unknown distribution (see equation (3.2)). It is a source of epistemic uncertainty.

- the spatial *distribution* of the equivalent heat source is not known with precision. We surmise that the main contribution to the overall uncertainty of the model is the spatial length-scale of the Goldak model. More details on the Goldak model are given in the previous chapter, but we remind that these parameters are :
 - a_f : the front half of the ellipsoid.
 - b : the half-width of the ellipsoid.
 - c : the depth of the ellipsoid.
 - η : the efficiency of the heat source.
 - K : the relationship between the rear and the front of the ellipsoid $\left(K = \frac{a_r}{a_f}\right)$.

Therefore, the parameter space $\mathcal{P} = \mathcal{M} \times \Theta$ space comprises two distinct blocks. The first block is formed by the set of known parameters $\mu = \{v, Q\} \in \mathcal{M}$, which will vary from one welding operation to the next, but can be controlled with precision. The other block is the set of unknown Goldak model parameters $\theta = \{a_f, b, c, \eta, K, \} \in \Theta$, which are endowed with a probability distribution $\theta \sim p_\theta$ that encodes the prior knowledge available about these parameters.

We remind that the unknown parameters are usually calibrated before simulating an experiment and that experts give their minimum and maximum values. If no further information is provided, p_θ is considered a uniform distribution between the extreme values.

There is one additional source of uncertainty, which does not depend on the parameters. Indeed, any measurement of the welding procedure is noisy, and this noise has to be modelled with a probability distribution whose parameters are directly determined by the sensor that generates the data.

3.1.2 “Truth” online Bayesian conditioning

Before introducing the inverse problem, let us remind that data will be assimilated at homogeneously distributed times $\bar{\mathcal{T}} = \{t^0 = 0, t^1 = \Delta t, t^2 = 2\Delta t, \dots, t^{N_t} = N_t \Delta t\}$. The assimilation time step Δt is adjusted so that the number of assimilation steps N_t is the same for all simulations, independently of the velocity v of the heat source,

$$\Delta t = \frac{(P_F - P_0)}{N_t v} \quad (3.1)$$

where P_F and P_0 are, respectively, the final and starting positions. In general, the position at time t is given by:

$$P(t) = P_0 + \left(\frac{tv}{(P_F - P_0)} - \epsilon_P(t) \right) (P_F - P_0) \quad (3.2)$$

Online at time $t^k \in \bar{\mathcal{T}}$, we assume that surface temperatures are measured noisily (i.e. with the infrared camera). Let us introduce N_H the nodes of the mesh that cover the surface observed by the infrared camera (see Chapter 2, Figure 2.9 for more details). The observed data, $d^k \in \mathbb{R}^{N_H}$ is written :

$$\mathbf{d}^k = H\mathbf{T}^k(\theta, \mu) + \boldsymbol{\epsilon}_M^k \quad (3.3)$$

where H is a fixed Boolean operator acting on the vector of finite element temperature nodal values \mathbf{T}^k at time step k , for the parameters μ and θ . The additive measurement noise is supposed to be zero-mean Gaussian distributed $\boldsymbol{\epsilon}_M \sim \mathcal{N}(0, \boldsymbol{\Sigma}_M = \sigma_M^2 I_{N_H})$. The sources of the temperature measurement uncertainty are varied and might include light reflections or lack of knowledge of material parameters like the emissivity at different temperatures, see [77] for a discussion on that topic. The measure error parameter σ_M^2 is calibrated empirically comparing the measure from two sensors available in EDF's welding lab, as shown in Chapter 2, Section 2.5.

The objective is to identify the unknown parameter values θ that best describe the observed data at a given time t^k , for a noisy observation \mathbf{d}^k :

$$p(\theta | \mathbf{d}^k, \mathbf{d}^{k-1}, \dots, \mathbf{d}^0, \mu) \quad (3.4)$$

where we consider a sequence of data.

Assuming that the successive noise vectors $\{\boldsymbol{\epsilon}^k\}_{k \in [0, \dots, N_t]}$ are independent, the statistical inverse problem to be solved online is the following:

At time $t^k \in \bar{\mathcal{T}}$, the posterior distribution of unknown parameters given the past measurements is

$$p(\theta | \mathbf{d}^k, \mathbf{d}^{k-1}, \dots, \mathbf{d}^0, \mu) \propto \Pi_{k' \leq k} \mathcal{L}^{k'}(\theta; \mathbf{d}^{k'}) p_\theta(\theta) \quad (3.5)$$

where $\mathcal{L}^{k'}(\theta; \mathbf{d}^{k'})$ is the likelihood function. The measurement error is considered Gaussian. The likelihood of the observation writes:

$$\forall k' \leq k \quad \mathcal{L}^{k'}(\theta; \mathbf{d}^{k'}) = \exp \left(-\frac{1}{2} (\mathbf{d}^{k'} - H\mathbf{T}^k(\theta, \mu))^T \boldsymbol{\Sigma}_M^{-1} (\mathbf{d}^{k'} - H\mathbf{T}^k(\theta, \mu)) \right) \quad (3.6)$$

A usual approach to compute a maximum a posteriori (MAP) estimator of the posterior distribution would be Markov Chain Monte-Carlo methods. This would require multiple evaluations of the Finite Elements model, which is not feasible in real-time. Using surrogate models to avoid the costly calls to the Finite Elements is, thus, necessary. However, one has to recall that the objective of the approach is not the calibration of the unknown parameters but the estimation of unobservable physical quantities such as stress. After calibrating the parameters, the following step would be the computation of the overall temperature field and the resolution of the mechanical problem up to t^k .

The method presented in the following sections aims at creating time-independent multiphysics local surrogate models that allow the estimation of observable and unobservable physical quantities and the calibration of the unknown parameters simultaneously and in real-time.

3.2 Local multiphysics PCA model

The previous section showed that the truth inverse problem is not well suited for real-time applications if solved with classical methods. In this section, in order to

overcome these issues, the construction of local surrogate models from snapshots of a Finite Elements parametric study is discussed. This approach relies on an offline-online procedure where local surrogate models are constructed from simulation data. Then, they are used to obtain real-time estimations of physical quantities and unknown model parameters.

In the first section, the notion of the joint state vector is introduced as well as the local positional surrogate models. Then, the theoretical resolution of the data assimilation is shown. The final section explains how to obtain the model parameters using a Singular Value Decomposition of snapshot matrices.

3.2.1 Local surrogate models

We propose to make local surrogate models for a linear joint representation of the state, known parameters, and unknown parameters for every position of the heat source at the different assimilation times t^k , where $k = 0, \dots, N_t$.

The positions are fixed in a grid between the initial position P_0 and the final position P_F with a regular separation ΔP . The parametric solutions are clustered by the position $P(t)$ of the heat source and associated to the position $P^k = k\Delta P$ if $|P(t) - P^k| < \frac{\Delta P}{2}$. In total, there are $\mathbb{P} = \lfloor \frac{P_F - P_0}{\Delta P} \rfloor$ positions.

The joint state vector at position P^k for a welding application is:

$$\mathbf{s}^k = \begin{pmatrix} \mathbf{T}^k \\ \boldsymbol{\sigma}_I^k \\ \boldsymbol{\mu}^k \\ \boldsymbol{\theta}^k \end{pmatrix} \quad (3.7)$$

where \mathbf{T}^k are the nodal values of temperature, $\boldsymbol{\sigma}_I^k$ are the nodal values of the maximum principal stress, $\boldsymbol{\mu}^k$ are the known parameters, and $\boldsymbol{\theta}^k$ the unknown parameters. The dimension of the joint state vector is very large and depends on the mesh used for the Finite Elements simulations. For a mesh containing N_n nodes, N_μ known parameters and N_θ unknown parameters, the dimension of \mathbf{s}^k is:

$$N_s = 2 \times N_n + N_\mu + N_\theta \quad (3.8)$$

The Finite Elements parametric study presented in the previous chapter uses a mesh with $N_n = 104101$ nodes. Let us underline that the known parameters are fixed for the duration of the experiment. From now on, we will write them as μ independently of the position.

The joint state vector will be expressed as a local reduced-order model. Local model approaches have been proposed to avoid the use of a single large reduced bases, for example in [128]. Usually, the partition is done by time, such as in [44]. We have chosen to use local models that capture the non-linear behaviour for all the parameter values at a single position in space. This partition of the heat source trajectory in discrete position points avoids the use of a single reduced basis that could be, potentially, too large for real-time applications and not precise enough for an accurate representation of the physical phenomena.

The local model at position P^k reads as

$$\mathbf{s}^k = \begin{pmatrix} \mathbf{T}^k \\ \sigma_I^k \\ \mu^k \\ \theta^k \end{pmatrix} = \bar{\mathbf{s}}^k + \Phi^k \boldsymbol{\alpha}^k + \boldsymbol{\epsilon}_T^k \quad (3.9)$$

with $\bar{\mathbf{s}}^k$ the mean of the state vector, $\boldsymbol{\alpha}^k \sim \mathcal{N}(0, \mathbf{I}_{n_\phi})$ and $\boldsymbol{\epsilon}_T^k \sim \mathcal{N}(0, \sigma_T^2 \mathbf{I}_{N_s})$. The operator $\Phi^k \in \mathbb{R}^{N_s \times N_{n_\phi}}$ -a decoder- is fixed for each position and possesses n_ϕ columns. It is obtained by using all parametric solutions of the welding problem corresponding to position P^k . This model provides a low-dimensional representation of the state \mathbf{s}^k in a subspace of dimension n_ϕ , with an error term $\boldsymbol{\epsilon}_T^k$.

This probabilistic model encodes the dependency between all the state variables and the known and unknown parameters in the form of a multivariate Gaussian:

$$\mathbf{s}^k \sim \mathcal{N}(\bar{\mathbf{s}}^k, \Sigma_{s^k}) \quad (3.10)$$

with $\Sigma_{s^k} = \Phi^k (\Phi^k)^T + \sigma_T^2 \mathbf{I}_{N_s}$

The probabilistic expression of the joint state vector defines a prior distribution. The solution of the inverse problem is now the probability distribution of \mathbf{s}^k given a set of known parameters μ and observed data $\mathbf{d}^k \in \mathbb{R}^{N_H}$: $p(\mathbf{s}^k | \mathbf{d}^k, \mu)$. The joint state vector contains different types of data, including various physical quantities (some of them might be unobservable, like stress in our application case) and unknown model parameters that need to be calibrated. All of them can be estimated at the same time using Bayes theorem. This is an essential advantage of these models. We can obtain quick estimations of the desired physical fields, and at the same time the unknown parameters of the Finite Elements model are calibrated, ensuring that a more detailed study of the experience can be launched without any further calibration. The application of Bayes theorem reads:

$$p(\mathbf{s}^k | \mathbf{d}^k, \mu) = \frac{p(\mathbf{d}^k | \mathbf{s}^k) p(\mathbf{s}^k)}{p(\mathbf{d}^k)} \quad (3.11)$$

The observed data, in a classical way, can be considered as a partial observation of the joint state vector with added measurement noise $\boldsymbol{\epsilon}_M^k$ that comes from the sensor, as presented in equation (3.3).

$$\mathbf{d}^k = H \mathbf{s}^k + \boldsymbol{\epsilon}_M^k \quad (3.12)$$

where $H \in \mathbb{R}^{N_H \times N_s}$ is a boolean operator that extracts the N_H observed nodes of the mesh and the known parameters from the multiphysics state vector. The effect of such a function is illustrated in future sections (for example, in Figure 3.1). The measurement error is the same for all frames of a video, so it is not dependent on k . The error is assumed to be Gaussian distributed:

$$\boldsymbol{\epsilon}_M \sim \mathcal{N}(0, \sigma_M^2 \mathbf{I}_{N_H}) \quad (3.13)$$

These Gaussian assumptions are particularly interesting for a real-time application. If the observed data is also supposed to have a Gaussian measurement error, it follows that it is also modelled by a Gaussian distribution:

$$\mathbf{d}^k \sim \mathcal{N}(\mathbf{H}\bar{\mathbf{s}}^k, \mathbf{H}\Sigma_{s^k}\mathbf{H}^T + \sigma_M^2\mathbf{I}_{N_H}) \quad (3.14)$$

With both \mathbf{s}^k and \mathbf{d}^k following Gaussian distributions, the posterior distribution $p(\mathbf{s}^k|\mu, \mathbf{d}^k)$ of the joint state vector will also be Gaussian and can be calculated analytically, significantly decreasing computation time. This avoids the need for expensive Markov Chain Monte Carlo methods to determine the posterior distribution. The mean of the posterior distribution gives the Maximum A Posteriori (MAP) estimator. The analytical expression of the mean and covariance is given by:

$$\mathbf{m}_{\mathbf{s}^k|\mathbf{d}^k, \mu} = \bar{\mathbf{s}}^k + \Sigma_{s^k}\mathbf{H}^T (\mathbf{H}\Sigma_{s^k}\mathbf{H}^T + \sigma_M^2\mathbf{I}_{N_H})^{-1} (\mathbf{d}^k - \bar{\mathbf{d}}^k) \quad (3.15)$$

$$\Sigma_{\mathbf{s}^k|\mathbf{d}^k, \mu} = \Sigma_{s^k} - \Sigma_{s^k}\mathbf{H}^T (\mathbf{H}\Sigma_{s^k}\mathbf{H}^T + \sigma_M^2\mathbf{I}_{N_H})^{-1} \mathbf{H}\Sigma_{s^k} \quad (3.16)$$

where $\bar{\mathbf{d}}^k$ is the mean of \mathbf{d}^k .

3.2.2 Maximum likelihood PCA

Two parameters of the local surrogate models presented in equation (3.9) need to be calibrated: the decoder Φ^k and σ_T^2 , that guides the error term.

Let us assume that a snapshot matrix is available for each heat source position $\mathbf{S}^k = \{\mathbf{s}_1^k, \mathbf{s}_2^k, \dots, \mathbf{s}_{n_p}^k\} \in \mathbb{R}^{N_s \times n_p}$, where each column of \mathbf{S}^k is composed of a joint state vector containing the objective fields, i.e. temperature and maximum principal stress for a welding application, and the set of known and unknown parameters at position P^k . There should be as many columns as the number of parameters in the Design of Experiment (DOE). Thus, the dimension of the snapshot matrices is $N_s \times n_p$. More details on the DOE and the computation of a numerical database of welding simulations were given in Chapter 2. Without loss of generality, we will drop the superindex k of the models and snapshot matrices for the rest of this section.

Instead of using a standard POD (or PCA) on the snapshot matrix, we will introduce the probabilistic PCA. In 1999, Tipping and Bishop[158] showed that a probabilistic formulation of PCA can be obtained from a Gaussian latent variable model, for which the basis vectors are maximum-likelihood estimates. This Gaussian latent variable model is described in Equation (3.9). Although all the parameters in Equation (3.9) can be computed with an Expectation Maximisation (EM) algorithm [158, 134], there are explicit expressions for the maximum likelihood estimates of the parameters when $\epsilon_T \sim \mathcal{N}(0, \sigma_T^2\mathbf{I}_{N_S})$. The Singular Value Decomposition of the snapshot matrix S reads:

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{W}^T \quad (3.17)$$

where $\mathbf{U} \in \mathbb{R}^{N_s \times n_p}$ is the matrix of left singular vectors, $\mathbf{\Lambda} \in \mathbb{R}^{n_p \times n_p}$ is a diagonal matrix containing the singular values and $\mathbf{W}^T \in \mathbb{R}^{n_p \times n_p}$ is the matrix of right singular values.

The maximum likelihood parameters are:

$$\Phi = \mathbf{U}_{n_\phi} \sqrt{(\mathbf{\Lambda}_{n_\phi} - \sigma^2\mathbf{I}_{n_\phi})} \mathbf{R} \quad (3.18)$$

where $\mathbf{U}_{n_\phi} \in \mathbb{R}^{N_S \times n_\phi}$ contains the left singular vectors associated to the greatest $n_\phi < n_p$ singular values of the snapshot matrix's SVD. The diagonal matrix $\mathbf{\Lambda}_{n_\phi}$ contains the greatest n_ϕ singular values in decreasing order. \mathbf{R} is an orthogonal rotation matrix that, in practice, will be chosen to be the identity matrix. Finally, σ_T^2 is related to the truncation error:

$$\sigma_T^2 = \frac{1}{n_p - n_\phi} \sum_{j=n_\phi+1}^{n_p} \lambda_j^2. \quad (3.19)$$

where λ_j are the diagonal elements of $\mathbf{\Lambda}$.

These results mean that the local probabilistic PCA models can be easily computed from an SVD of the output of a parametric study of the finite thermo-mechanical elements model.

3.2.3 Offline stage methodology

To sum up this section on the local multiphysics PCA models, we will now list the different steps of the offline stage :

Offline methodology

- 1 Choose a fixed assimilation grid from which observation data will be acquired.
- 2 Cluster the parametric Finite Elements solutions by heat source positions.
- 3 Compute an SVD for each position.
- 4 Choose n_ϕ (the number of modes in the reduced basis) and compute the Probabilistic PCA.

Let us remind that the two error terms ϵ_T and ϵ_M are modelled by two parameters, σ_T and σ_M . The value of σ_T is given by the Maximum-Likelihood estimator (see Equation (3.19)), whilst the measurement error depends on the sensors used to obtained the experimental data. The error of the infrared image was estimated in Chapter 2.

3.3 Online prediction in real time

Once the local models are constructed, the offline stage is finished. Now, we need to consider the online application of the local models. The input data is a vector of known parameters and an infrared image of the experiment. The two problems to solve are the quick computation of the mean and covariance of the posterior distribution presented in equations (3.15) and (3.16) and the selection of the best local model for the input data.

The large dimension of the joint state vector and the reduced bases are the main obstacles towards a real-time application. High fidelity simulations involve large and detailed meshes, leading to very high-dimensional data.

The selection of the best local model is of great importance. Using the incorrect model will not yield accurate estimations because the encoded data in the multi-physics bases is position-dependent. Two cases have to be considered depending on whether the torch position is known or not. The first case, when the heat source position is known at all times, is straightforward, and the predictions are obtained using a single well-identified PPCA model. In the second case, the torch position needs to be estimated with some uncertainty. The predictions will be computed from a mixture of local PPCA models.

In the first subsection, we will tackle the computation time issue, detailing how to rewrite equations (3.15) and (3.16) for real-time applications. The next two sections will focus on the online selection of local models, separating the cases where the torch position is known and unknown. In particular, we will see how to estimate the heat source position from an infrared image and deal with the estimation's uncertainty.

3.3.1 Fast computation of mean and covariance

The evaluation of equations (3.15) and (3.16) could be potentially slow due to the dimensions of the matrices involved. To illustrate this issue, we remind that the mesh used to generate the simulation database contains $N_n = 104101$ nodes. A joint state vector with two different physical quantities has a dimension of $N_S = 2 \times N_n + N_\mu + N_\theta = 208209$. The reduced bases $\Phi^k \in \mathbb{R}^{N_S \times n_\phi}$ are also large.

To avoid this cost, we propose to compute the posterior distribution of the reduced coordinates α^k instead. The prior distribution is Gaussian $\alpha^k \sim \mathcal{N}(0, I_{n_\phi})$, so the posterior distribution $p(\alpha^k | d^k, \mu)$ is also Gaussian. The mean and covariance of the posterior distribution have the following explicit expression:

$$m_{\alpha^k | d^k, \mu} = (\Phi^k)^T H^T \left(H \Phi^k (\Phi^k)^T H^T + (\sigma_M^2 + \sigma_T^2) \mathbf{I}_{N_H} \right)^{-1} (d^k - \bar{d}^k) \quad (3.20)$$

$$\Sigma_{\alpha^k | d^k, \mu} = \mathbf{I}_{n_\phi} - (\Phi^k)^T H^T \left(H \Phi^k (\Phi^k)^T H^T + (\sigma_M^2 + \sigma_T^2) \mathbf{I}_{N_H} \right)^{-1} H \Phi^k \quad (3.21)$$

where $\Phi^k \in \mathbb{R}^{N_S \times n_\phi}$ is the PPCA basis, $H \in \mathbb{R}^{N_H \times N_S}$ is the Boolean observation function, $d^k \in \mathbb{R}^{N_H}$ is the observed data, N_H is the dimension of the observed data and σ_M and σ_T are the parameters guiding the measurement and truncation errors.

The mean of the posterior distribution of s^k can be then deduced using the reduced basis Φ^k as follows:

$$m_{s^k | d^k, \mu} = \bar{s}^k + \Phi^k m_{\alpha^k | d^k, \mu} \quad (3.22)$$

As for the covariance matrix, it can be calculated by:

$$\Sigma_{s^k | d^k, \mu} = \Phi^k \Sigma_{\alpha^k | d^k, \mu} (\Phi^k)^T + \sigma_T^2 \mathbf{I}_{N_S} \quad (3.23)$$

The whole covariance matrix might be impossible to store in RAM if the number of degrees of freedom is large. For instance, in our application case, the covariance matrix would be a full matrix of dimensions $N_S \times N_S = 208209 \times 208209$. In this case, only the diagonal of the matrix product $\Phi^k \Sigma_{\alpha^k | d^k, \mu} (\Phi^k)^T$ is calculated. This would represent, for each node of the mesh, the variance of its associated value of temperature and stress.

Remark

The covariance matrix is a full matrix of dimension $N_S \times N_S = 208209 \times 208209$. Such a matrix needs an amount of memory that cannot be allocated. Only the diagonal of the covariance matrix is calculated.

This first change improves the computation times, but there are still parts of the equation that are time-consuming for very large matrices. Further algebraic manipulation is needed for computation efficiency. Let's introduce the notation $\mathbf{X} := H\Phi^k \in \mathbb{R}^{N_H \times n_\phi}$ and $\mathbf{Z} := (\sigma_M^2 + \sigma_T^2)\mathbf{I}_{N_H} \in \mathbb{R}^{N_H \times N_H}$ to simplify the reading of the expressions. Equations (3.20) and (3.21) become

$$\mathbf{m}_{\alpha^k|d^k,\mu} = \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \mathbf{Z})^{-1} (\mathbf{d}^k - \bar{\mathbf{d}}^k) \quad (3.24)$$

$$\Sigma_{\alpha^k|d^k,\mu} = \mathbf{I}_{n_\phi} - \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \mathbf{Z})^{-1} \mathbf{X} \quad (3.25)$$

Mean of the posterior Gaussian distribution

To accelerate the evaluation, this algebraic identity, found in Appendix C of [23], is used:

$$(\mathbf{P}^{-1} + \mathbf{B}^T \mathbf{R}^{-1} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{R}^{-1} = \mathbf{P} \mathbf{B}^T (\mathbf{B} \mathbf{P} \mathbf{B}^T + \mathbf{R})^{-1} \quad (3.26)$$

These matrices have the following dimensionalities: $\mathbf{P} \in \mathbb{R}^{M \times M}$, $\mathbf{R} \in \mathbb{R}^{N \times N}$ and $\mathbf{B} \in \mathbb{R}^{N \times M}$. Equality (3.26) can be verified by multiplying both sides by $(\mathbf{B} \mathbf{P} \mathbf{B}^T + \mathbf{R})$.

Interestingly, this expression is such that depending on the relationship between N and M, one side of it is quicker to evaluate than the other. The left side of the equation is faster when $M \ll N$. On the contrary, if $N \ll M$, the right side is cheaper.

It is easy to see that $\mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \mathbf{Z})^{-1}$ in equation (3.24) corresponds to the right side of equation (3.26) when $\mathbf{P} = \mathbf{I}_{n_\phi}$, $\mathbf{B} = \mathbf{X}$ and $\mathbf{R} = \mathbf{Z}$. In our case, the left side of Eq. (3.26) is faster because n_ϕ , the number of modes in the PPCA basis, is always smaller than N_H , the number of mesh nodes observed by the infrared camera. For a faster implementation, the expressions that should be evaluated are:

$$\mathbf{m}_{\alpha^k|d^k,\mu} = (\mathbf{X}^T \mathbf{Z}^{-1} \mathbf{X} + \mathbf{I}_{n_\phi})^{-1} \mathbf{X}^T \mathbf{Z}^{-1} (\mathbf{d}^k - \bar{\mathbf{d}}^k) \quad (3.27)$$

$$\Sigma_{\alpha^k|d^k,\mu} = \mathbf{I}_{n_\phi} - (\mathbf{X}^T \mathbf{Z}^{-1} \mathbf{X} + \mathbf{I}_{n_\phi})^{-1} \mathbf{X}^T \mathbf{Z}^{-1} \mathbf{X} \quad (3.28)$$

Notice that \mathbf{Z} is a diagonal matrix, which means that computing \mathbf{Z}^{-1} is trivial, and $\mathbf{X}^T \mathbf{Z}^{-1}$ is very cheap.

Covariance of the posterior Gaussian distribution

In the case of the covariance matrix, further improvements are possible using the Woodbury identity[70] rather than the previous transformation. The Woodbury identity is :

$$(\mathbf{A} + \mathbf{UBV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{B}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1} \quad (3.29)$$

where $\mathbf{A} \in \mathbb{R}^{M \times M}$, $\mathbf{U} \in \mathbb{R}^{M \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times N}$ and $\mathbf{V} \in \mathbb{R}^{N \times M}$.

If $\mathbf{A} = \mathbf{I}_{n_\phi}$, $\mathbf{U} = \mathbf{X}^T$, $\mathbf{B} = \mathbf{Z}^{-1}$ and $\mathbf{V} = \mathbf{X}$, then we recognize that the right side of equation (3.29) is the original expression of the posterior covariance matrix in equation (3.25). Writing the covariance as in the left-hand side of the Woodbury identity leads to:

$$\Sigma_{\alpha^k|d^k,\mu} = (\mathbf{X}^T \mathbf{Z}^{-1} \mathbf{X} + \mathbf{I}_{n_\phi})^{-1} \quad (3.30)$$

Notice that equation (3.30) is part of equation (3.27), meaning that it only needs to be computed once, which saves a lot of computation time.

Example

Let us provide two examples of the computation time needed to evaluate the old expressions in Equations (3.24) and (3.25), and the new ones in Equations (3.27) and (3.30). The first example uses input on all nodes on the surface of the specimen ($N_H = 8093$) and 20 PPCA modes ($n_\phi = 20$). The second example uses input on a limited area seen by the infrared camera (the surface shown in Figure 3.6, with $N_H = 3418$) and 20 PPCA modes ($n_\phi = 20$).

The results shown in Table 3.1 are the average of 7000 runs for both examples using each of the expressions presented previously. The computation time is greatly reduced using the expression in equations (3.27) and (3.28), with speedup factors of 411 and 16 respectively.

TABLE 3.1: Computation time with both expressions with two different observation functions.

	Old expression	New expression	Speedup
Whole surface	1.55 s	0.00377 s	~ 411
Camera area	0.337 s	0.00212 s	~ 16

3.3.2 Known position

Let us assume that the heat source position $P(t)$ is known at all times. The selection of the best local model is trivial: it is the one located closer to the heat source.

Perfect knowledge of the heat source position is not always assured. This situation arises, for example, when the robotic arm is equipped with a sensor that measures the travelled distance. Another possibility is that all measurements are synchronised. Knowing how much time has passed between the start of the experiment and the current time is enough to determine precisely the heat source position $P(t)$ in an infrared image because the travel speed is a known parameter. However, even in these cases, there might be sources of uncertainty, such as noise in the sensor signals.

3.3.3 Unknown position

As is the case in EDF's welding lab, we may not have access to the exact heat source position because of a lack of synchronicity between the measurement sensors. In this case, $\epsilon_P(t) \neq 0$ in equation (3.2) and the torch position needs to be estimated from the video frame, which adds more uncertainty to the model. The filmed side of the specimen is the opposite side of the weld. There exists a delay between the highest temperature on this side and the torch position. Furthermore, this delay is dependent on material and operational parameters, such as speed.

Using the finite elements simulations presented in the previous chapter, we created a Gaussian surrogate model that links the position of the highest temperature measured on the camera side of the specimen \mathbf{x} and the known parameters μ with the position of the heat source on the welded surface of the specimen P^k . The surrogate model reads as:

$$P^k = f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (3.31)$$

where $\mathbf{x} := (x^k, \mu)$ is a vector containing the highest temperature position and the known operational parameters. The Gaussian Process is entirely determined by its mean, given by the deterministic function $m(\mathbf{x})$, and its covariance, given by the function $k(\mathbf{x}, \mathbf{x}')$. The model is fitted using a Matérn kernel [57, 126] as covariance function:

$$k(\mathbf{x}, \mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} d(\mathbf{x}, \mathbf{x}') \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{l} d(\mathbf{x}, \mathbf{x}') \right), \quad (3.32)$$

In equation (3.32), $\Gamma(\cdot)$, $d(\cdot, \cdot)$ and $K_\nu(\cdot)$ are, respectively, the gamma function, the euclidean distance and a modified Bessel function. The parameters $\nu > 0$ and $l > 0$ were fixed to $\nu = \frac{3}{2}$ and $l = 1$. This choice of ν ensures that the process f is differentiable[57, 126].

Given a set of observations $\mathcal{P} = \{(\mathbf{x}_1, P^1), \dots, (\mathbf{x}_n, P^n)\}$, the posterior distribution $p(P^k | \mathcal{P})$ can be derived. Thus, the posterior distribution represents a family of functions that link \mathbf{x} and a position of the heat source. We could use the mean of the posterior distribution and evaluate it for any \mathbf{x} . However, in the case where the posterior distribution's variance is larger than the separation ΔP between two consecutive positions in the assimilation grid, assigning a local model remains uncertain.

Indeed, we decided to sample the posterior distribution $M = 1000$ times and evaluate each function for \mathbf{x} . In practice, this gives $M = 1000$ candidate positions of the heat source. Individually, each candidate positions \hat{P}_i , for $1 \leq i \leq M$, is associated with a PPCA model, determined by its reduced basis Φ^i . We define an empirical discrete probability distribution of assignment given by:

$$\forall j \in \llbracket 0, N_t \rrbracket \quad p_j = \frac{1}{M} \sum_{i=1}^M \mathbb{1}_{\{\hat{P}_i \in [P^j - \frac{\Delta P}{2}, P^j + \frac{\Delta P}{2}]\}} \quad (3.33)$$

Thus, the probability of assignment of a local model is 0 if not a single one of the M candidate positions is associated with it. We define J as the set of indices of the active local models:

$$J = \{j \in \llbracket 0, N_t \rrbracket : p_j > 0\} \quad (3.34)$$

The multiphysics state we want to estimate, \tilde{s}^k is calculated as a Gaussian mixture where the probabilities p_j are the weights of each local model. The mean and covariance of the posterior distribution are computed as the weighted sum of each active model's mean and covariance posterior distribution. Naming the observed data of the unknown position d^k :

$$\mathbf{m}_{s^k|d^k,\mu} = \sum_{j \in J} p_j \mathbf{m}_{s^j|d^k,\mu} \quad (3.35)$$

$$\Sigma_{s^k|d^k,\mu} = \sum_{j \in J} p_j \Sigma_{s^j|d^k,\mu} \quad (3.36)$$

were $\mathbf{m}_{s^j|d^k,\mu}$ and $\Sigma_{s^j|d^k,\mu}$ are the mean and covariance of the posterior distribution using the local model located at position P^j , $\forall j \in J$.

The torch position estimation is sufficiently accurate to ensure that the number of active PPCA models is not too large, as we will see in a future section. The assimilation is still performed in real-time, thanks to the rapid evaluation of each Bayesian inference.

3.3.4 Online stage methodology

To sum up this section on the online use of the local surrogate models, we will list the consecutive steps, considering the known and unknown torch position dichotomy:

Online stage methodology

1. We obtain a new infrared image.
2. If the torch position is known:
 - (a) Find the closest position P^k in the assimilation grid and select the associated local model.
 - (b) Compute the Maximum A Posteriori estimation of the state \mathbf{s} : $p(\hat{\mathbf{s}}|d^k, \mu)$.
3. If the torch position is unknown:
 - (a) Find the heat source position posterior distribution using the kriging surrogate model (Equations (3.31) and (3.32)).
 - (b) Sample the posterior distribution and associate a local model to each predicted position.
 - (c) Compute the discrete probability weights for each position.
 - (d) Assign weights to each model induced by J .
 - (e) Compute the MAP from a mixture of the active local models.

3.4 Forecasting of future states

Despite the lack of Markovian assumption in the models, they can forecast future states without observed experimental data. Assuming that the unknown parameters are not changed online, which is the case in a welding application, previously estimated values can be used to obtain predictions of future states.

Let us suppose that we have previously estimated a joint state vector $\hat{\mathbf{s}}^k$, including a set of unknown parameters $\hat{\theta}^k$, with the PPCA model at position P^k and that predictions of a future state are desired. The position of the future state is defined by a displacement of $\delta_z = z\Delta P$ in the direction of the weld, where ΔP is the regular separator in the positions grid.

Again, this opens two possibilities whether the torch position was known or was estimated from the data frame. If the heat source position was known, the prediction is computed using the PPCA model for position $P^{k+z} = (k+z)\Delta P$. Taking the estimated unknown parameters θ^k as an observation implies that the observation function H is restricted to a function that observes μ and $\hat{\theta}^k$ in the multiphysics state vector. The posterior distribution in this case is:

$$p(\tilde{\mathbf{s}}^{k+z}|\mu, \theta^k) \quad (3.37)$$

where $\tilde{\mathbf{s}}$ indicates that the joint state vector is forecasted. Notice that computing this posterior distribution is done with Equation (3.20) with \mathbf{d}^k is a vector that contains μ and $\hat{\theta}^k$ and σ_M^2 is the posterior covariance of the estimated unknown parameters.

The other possibility is that the current position was estimated from the experimental data. In this case, the displacement will be added to the position samples to maintain the torch position's uncertainty. This means that δ_z is added to each candidate position P_i , with $1 \leq i \leq M$:

$$\hat{P}_i = P_i + \delta_z, \quad \forall 1 \leq i \leq M \quad (3.38)$$

The local basis assignment changes, and a new set J is defined for this problem. Then, the prediction will be calculated using the mixture of PPCAs described in the previous section. Remark that, if the position was unknown, (3.37) becomes

$$p(\hat{\mathbf{s}}^{\hat{k}+j}|\mu, \hat{\theta}^{\hat{k}}) \quad (3.39)$$

where $\hat{\cdot}$ indicates that the position is unknown. In particular, $\tilde{\mathbf{s}}^{\hat{k}}$ is estimated from a mixture of local PPCAs. The mean and posterior distribution are now computed by:

$$\mathbf{m}_{\tilde{\mathbf{s}}^{\hat{k}}|\mu, \hat{\theta}^{\hat{k}}} = \sum_{j \in J} p_j \mathbf{m}_{\tilde{\mathbf{s}}^j|\mu, \hat{\theta}^{\hat{k}}} \quad (3.40)$$

$$\Sigma_{\tilde{\mathbf{s}}^{\hat{k}}|\mu, \hat{\theta}^{\hat{k}}} = \sum_{j \in J} p_j \Sigma_{\tilde{\mathbf{s}}^j|\mu, \hat{\theta}^{\hat{k}}} \quad (3.41)$$

3.4.1 Forecasting methodology

The following list details the steps to perform a forecast of a future state:

Online stage methodology

1. Compute an estimation of the joint multiphysics state \mathbf{s}^k .
2. If the torch position is known:
 - (a) Estimate the joint multiphysics state at position P^{k+z} using its corresponding local model by $p(\tilde{\mathbf{s}}^{k+z}|\mu, \hat{\theta}^k)$.
3. If the torch position is unknown:
 - (a) Find the heat source position posterior distribution using the kriging surrogate model.
 - (b) Sample the posterior distribution and add δ_z to the candidate positions.
 - (c) Associate a local model to each position.
 - (d) Assign weights to each model.
 - (e) Compute the MAP from a mixture of the active local models using μ and $\hat{\theta}^k$ as input data.

3.5 Numerical results

The following numerical tests use two different data sources to showcase the previously mentioned situations: the torch position is well known, and an estimation of the heat source position is needed. The torch position will be deduced from the infrared image and the state estimation computed from a mixture of PPCAs. The first test will use noisy synthetic data obtained from a previously calibrated Finite Element solution as input. The joint multiphysics state and unknown parameters estimation is obtained using a single PPCA model located at the exact position. Then, we will consider a frame of an infrared video of a real PVR experiment, in particular the experiment from the 2019 experimental campaign presented in Chapter 2. Lastly, the forecasting capabilities of the model will be shown. Considering the real experimental case as reference, we will predict a state situated 12 mm further.

As a reminder, none of the parameters used for the experiments was part of the Finite Elements parametrical study.

3.5.1 Estimation of the heat source position

The first step in the online stage is to identify the heat source position and its associated local models. A frame of the infrared video, out of 450, is selected arbitrarily for these tests. The projection of the infrared video frame on the FE mesh is shown in Figure 3.1.

The heat source position is estimated using the surrogate model presented in section 3.3.3. First, the highest temperature position is identified. We use this temperature and the 2019 experiment's welding parameters μ ($v = 2.0 \text{ mm/s}$ and $Q = 680.4 \text{ W}$, as indicated in Chapter 2) to obtain the posterior distribution of

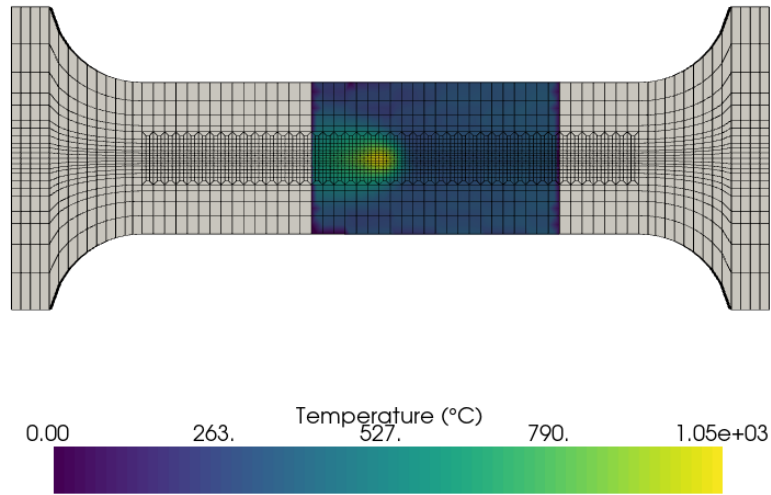


FIGURE 3.1: Projection of infrared images on the FE mesh.

the heat source position. We sample $M = 1000$ torch candidate positions from this posterior distribution and find their associated models. Figure 3.2 shows the discrete empirical probability of assignment of each local model. The most probable model is the one located at position 130, which corresponds to $65mm$ from the starting position. This estimation is relatively close to the one assessed by the equipment. Only seven models are active in the mixture of PPCAs, and their weight is given by the distribution in Figure 3.2.

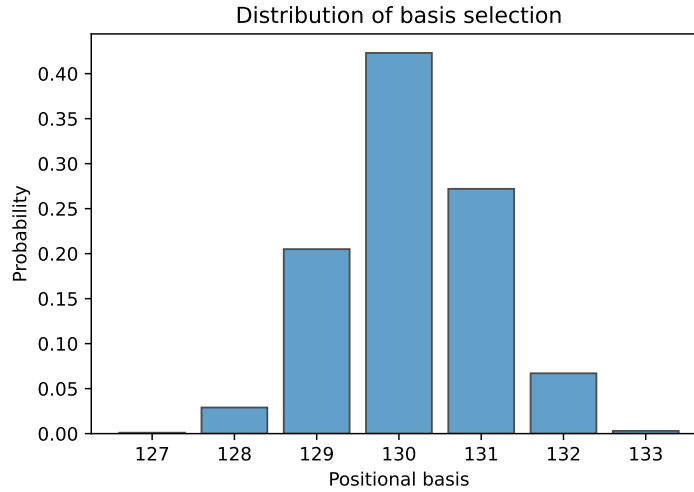


FIGURE 3.2: Possible PPCA bases with associated probability.

The snapshot for position 130 will be used as input for the noisy synthetic data case.

3.5.2 Tests with noisy synthetic data

In this first test, the input data comes from the calibrated finite elements simulation of the experiment of the 2019 experimental campaign, which uses $\mu = (v = 2.0 \text{ mm/s}, Q = 680.4 \text{ W})$. This experimental configuration is the same as the nominal energy experiments of the 2021 experimental campaign. As indicated previously, the snapshot corresponds to the position with the highest probability. White noise of the same amplitude as the measurement error ϵ_M , estimated for the camera, has been added to the data. The measurement error ϵ_M is calculated as described in Chapter 2, Section 2.5. The observation function H restricts the view to a region "seen" by the camera, and it is represented in Figure 3.3.

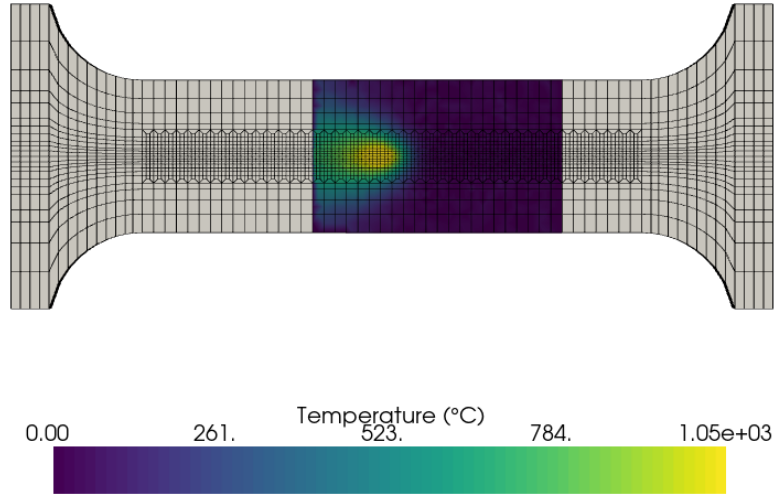


FIGURE 3.3: Noisy simulation data on camera area.

The mean of the posterior distribution $\mathbf{m}_{s^k|d^k}^k$ contains the estimation of the temperature and maximum principal stress fields, as well as the unknown parameters. We can compare these results to the calibrated finite elements simulation, from which the input data was taken. First, we can calculate a global relative reconstruction error between the mean posterior fields and the Finite Elements simulations. This relative error is calculated as:

$$e_T^k = \frac{\|\mathcal{T}(\mathbf{m}_{s^k|d^k,\mu}^k) - \mathbf{T}_{|\mu,\theta}^k\|_2}{\|\mathbf{T}_{|\mu,\theta}^k\|_2} \quad (3.42)$$

$$e_{\sigma_I}^k = \frac{\|\sigma_{\mathcal{I}}(\mathbf{m}_{s^k|d^k,\mu}^k) - \boldsymbol{\sigma}_{I|\mu,\theta}^k\|_2}{\|\boldsymbol{\sigma}_{I|\mu,\theta}^k\|_2} \quad (3.43)$$

where $\mathbf{T}_{|\mu,\theta}^k$ and $\boldsymbol{\sigma}_{I|\mu,\theta}^k$ are the nodal temperature and principal stress simulation values, $\mathcal{T}(\mathbf{m}_{s^k|d^k,\mu}^k)$ and $\sigma_{\mathcal{I}}(\mathbf{m}_{s^k|d^k,\mu}^k)$ are the estimated nodal temperature and principal stress extracted from the joint state vector, and $\|\cdot\|_2$ is the euclidean norm.

The 3D reconstruction of the temperature and stress fields is very accurate, with a global relative error of 0.489 % for the temperature and 6.857 % for the principal stress.

For visualisation reasons, the following results are compared only along a line of 130 mm at the centre of both sides of the specimen, which coincides with the weld line on the torch side. The 95% confidence interval of the estimation is also plotted around the posterior mean in Figure 3.4 and Figure 3.5.

The confidence intervals are thin for the observed temperature data, as is expected, but it is larger around the peak on the non-observed surface. Although no data was seen for the principal stress estimations, the estimated posterior mean is close to the simulation on both surfaces. One can also notice that the principal stress estimation is closer to the FE field on the camera side compared to the torch side. This illustrates the fact that observed temperature has also an impact on the principal stress accuracy.

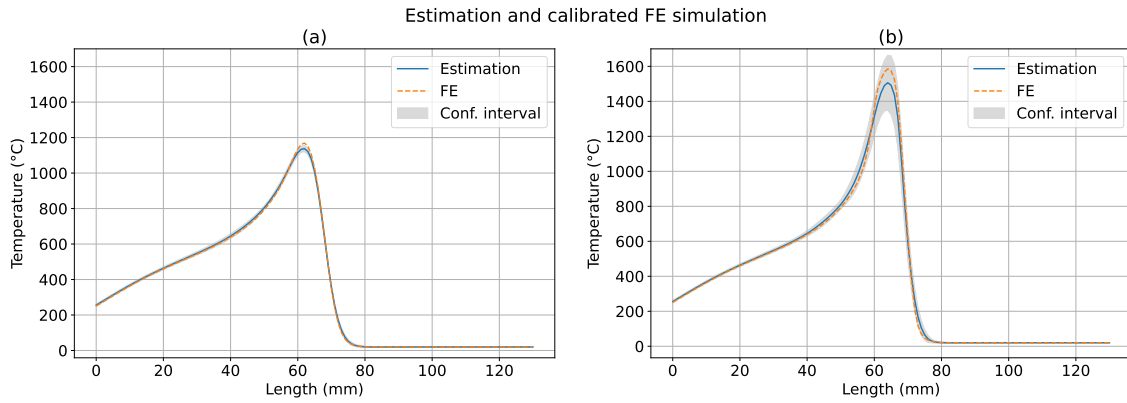


FIGURE 3.4: Temperature estimations, FE simulation and confidence interval. **(a)** Camera side (observed noisy FE data). **(b)** Torch side (No observed data).

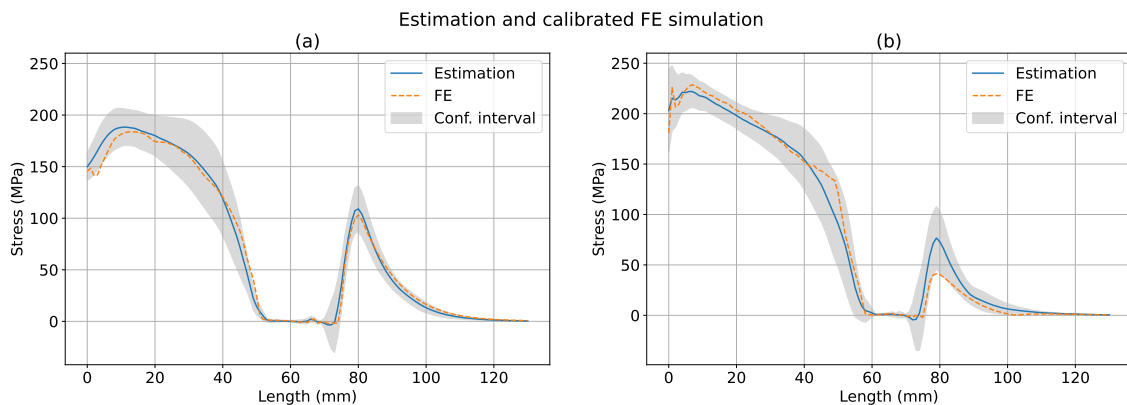


FIGURE 3.5: Maximum principal stress estimations, FE simulation and confidence interval. **(a)** Camera side (observed noisy FE data). **(b)** Torch side (No observed data).

The estimation of the unknown parameters θ is very close to the results of the deterministic calibration presented in the previous chapter. A comparison of the values obtained by the deterministic calibration and the mean posterior theta estimation is given in Table 3.2. It also shows the relative error between the values calibrated with thermocouple temperature data and the MAP of the posterior distribution estimated from the noisy snapshot.

TABLE 3.2: Posterior estimation of the unknown parameters θ .

	a_f	b	c	η	K
Calibrated value	6.657	3	1.5	0.9	1.15
Estimated value	6.189	2.842	1.586	0.890	1.165
Relative error	7.03 %	5.27 %	5.73 %	1.11 %	1.30 %

The computation time needed to produce the multiphysics state estimation is 3.04 milliseconds. Calculating the covariance matrix is more expensive, at 34.76 milliseconds, in particular the expression $\Phi^k \Sigma_{\alpha^k|d^k,\mu} (\Phi^k)^T$ is very costly despite only calculating the diagonal. The total computation time of the posterior distribution is 0.038 seconds. These computation times are the average results of 1000 runs.

3.5.3 Tests with real experimental data

In this test, we will use the camera frame projected onto the mesh as input data for the model. The camera is configured to capture temperatures above 400°C , so the observation function H is modified to only use data above 400°C , which is the area represented in Figure 3.6.

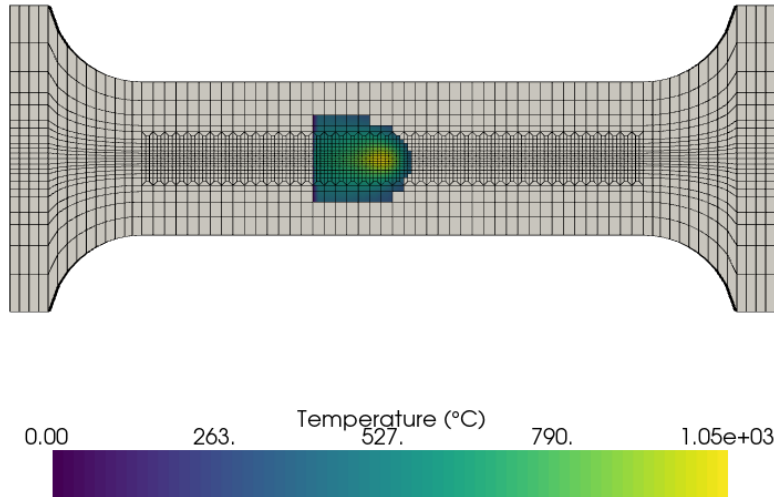


FIGURE 3.6: Experimental data - Temperatures above 400°C

The estimation is calculated as a mixture of the results given by the local PCA models in Figure 3.2. The amount of active local models is small, and each individual

conditioning is computed very fast, amounting to a total of 0.256 seconds for the entire posterior distribution estimation. The multiphysics state can be computed in only 0.019 seconds. As we observed in the previous case, computing the diagonal of the covariance matrix is more expensive, with a computation time of 0.237 seconds. Both computation times are the average of 1000 runs. All the confidence intervals are the 95% confidence intervals.

TABLE 3.3: Posterior estimation of the unknown parameters θ .

	a_f	b	c	η	K
Calibrated value	6.657	3	1.5	0.9	1.15
Estimated value	6.408	3.855	1.47	0.736	1.098

The estimations can be compared to the measured temperature on the camera side. In Figure 3.7, we can see that the model estimates a temperature that follows the experimentally measured one. The relative error between the measured temperature and the estimated one is 2.85 %. In Figure 3.8, the state estimation is compared to the FE simulated temperature field. We observe that the estimation deviates from the simulation. Indeed, the temperature is higher for the FE results. We interpret this difference in the estimation and the FE simulation as a model correction given by the partial observation of the temperature. We remind that the FE model was calibrated using only the thermocouple measures and not the infrared camera. This difference between the FE simulation and the measured data in Figure 3.8 may indicate that the calibration of θ using the thermocouples does not ensure a total agreement between the simulation's thermal dynamics and temperature observations. This is supported by the fact that the estimated efficiency η , shown in Table 3.3 along with the other unknown parameters, is smaller at around 0.74 while the initial calibration estimated it at 0.9. A smaller efficiency would transfer less heat to the specimen, explaining the lower estimated temperature obtained with the local PPCA model.

It can be noticed that on the camera side, the confidence interval is very small for the temperature estimation, as it is expected from observed data, whilst on the torch side, the confidence interval is larger, especially around the peak position. The maximum principal stress confidence intervals are relatively large, as no mechanical data is observed. There is an area where this is an exception, around the current heat source position, where there is no variance. This is explained by the physics of the problem. Indeed, there are no stresses at the heat source position in any of the parametric FE simulations.

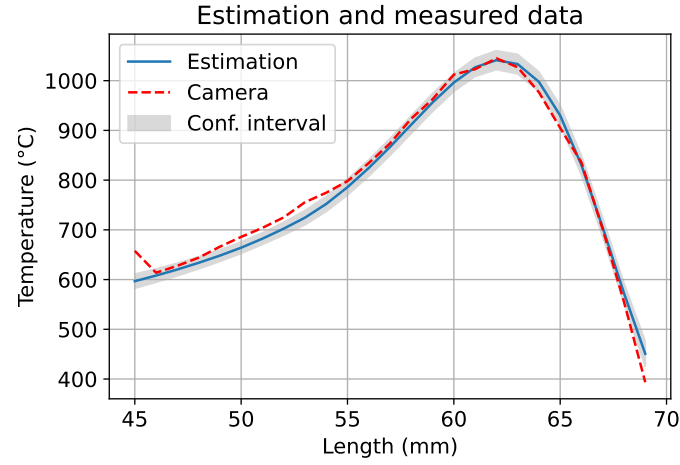


FIGURE 3.7: Temperature estimations and camera data.

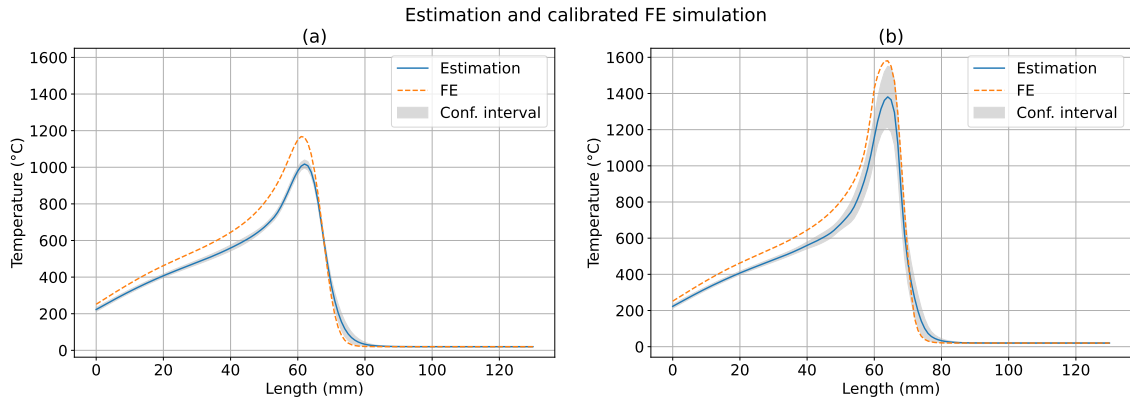


FIGURE 3.8: Temperature estimations, FE simulation and confidence interval. (a) Camera side. (b) Torch side.

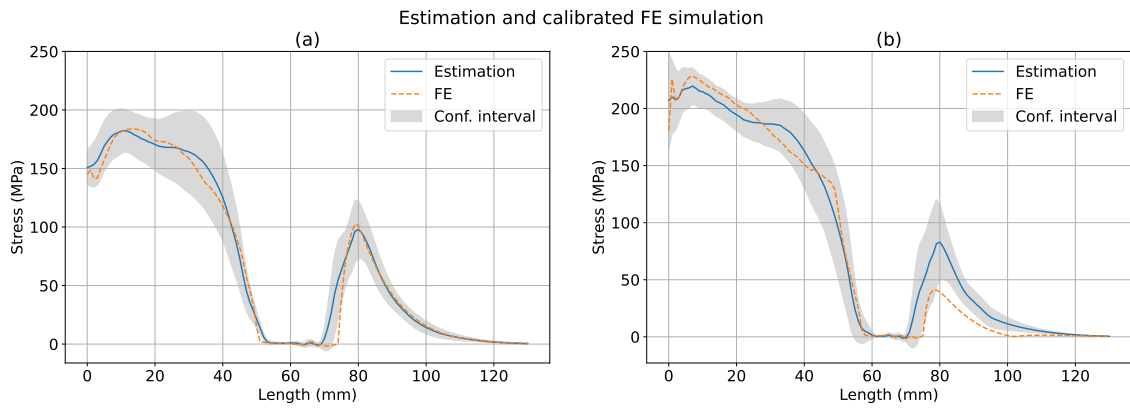


FIGURE 3.9: Maximum principal stress estimation, FE simulation and confidence interval. (a) Camera side (observed infrared camera data). (b) Torch side (no observed data).

Forecasting

In this last example, the forecasting capabilities of the model will be shown. Let us assume that we want to estimate the temperature and maximum principal stress fields in a future position that has not been observed yet. This position is situated 12 mm further than the one studied for the previous tests, which was shown in Figure 3.1. The new frame is shown in Figure 3.10. We are still in the case where the torch position is unknown. The future position is represented by a frame corresponding to an advancement of 12 mm from the previous one, thus. All the information available is the value of the known parameters $\mu = (v = 2.0 \text{ mm/s}, Q = 680.4 \text{ W})$ and the posterior estimation of the unknown parameters with their posterior covariance. The estimated values are displayed on the second row of Table 3.3.

Choosing the active coefficient of the mixture of PPCAs for this position $P^j = (k + 12)\Delta P$ is the first problem to solve. When an image is available, the surrogate model for position estimation takes the highest temperature position identified in the image and returns the probability distribution of the torch position. In the forecasting problem, we have no video frame to find the highest temperature, so the probability distribution obtained previously (see Figure 3.2) is shifted by the 12 mm. It is then used to obtain the discrete assignment probability. The multiphysics state \tilde{s}^j will be predicted by conditioning its distribution by the known parameters μ and the estimated unknown parameters $\theta^{\hat{k}}$ with its associated posterior variance, following equation (3.41).

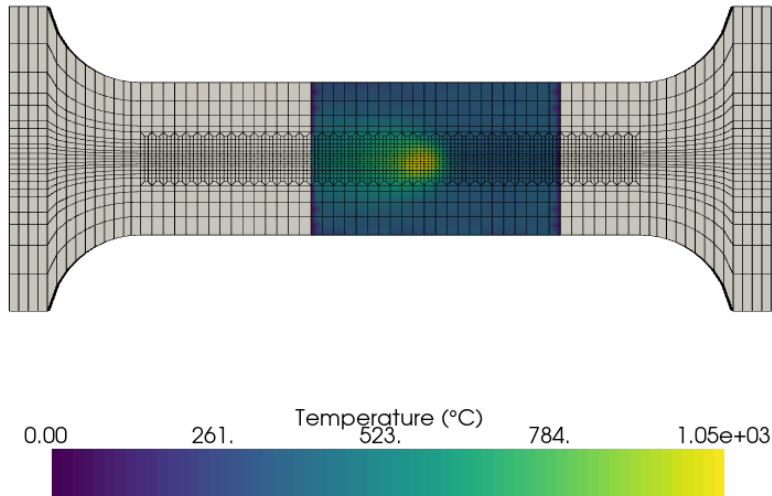


FIGURE 3.10: Camera data of the future state.

As with previous tests, the predicted estimations, $p(\tilde{s}^j | \mu, \theta^k)$ are compared to the experimental data and a Finite Elements simulation over a line on both sides of the specimen. The infrared camera measurements corresponding to the centre line on Figure 3.11 are compared to the predictions in Figure 3.11. The estimation is very close to the camera data, but the 95% confidence interval is very large compared to the one in Figure 3.7, a case where the data was observed, thus, reducing the uncertainty.

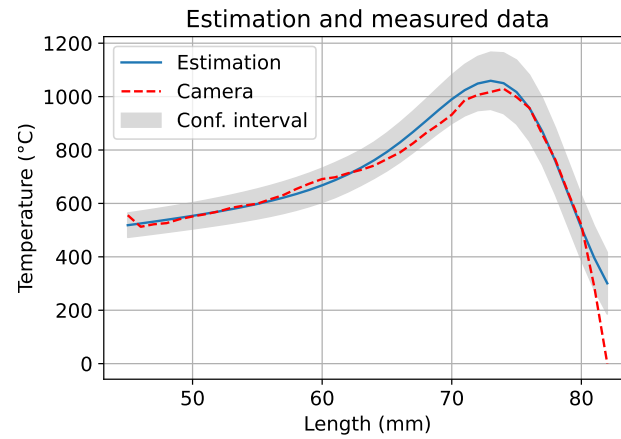


FIGURE 3.11: Forecasted temperature and camera data.

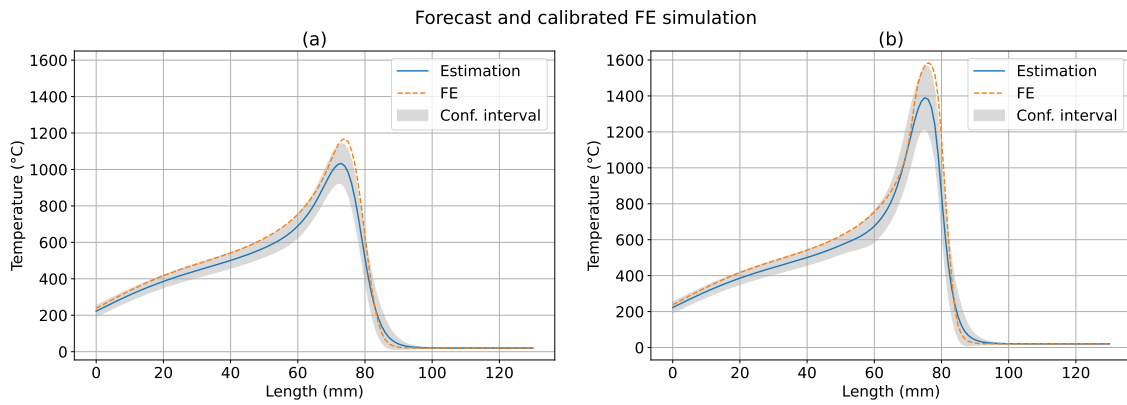


FIGURE 3.12: Forecasted temperature, FE simulation and confidence interval. (a) Camera side. (b) Torch side.

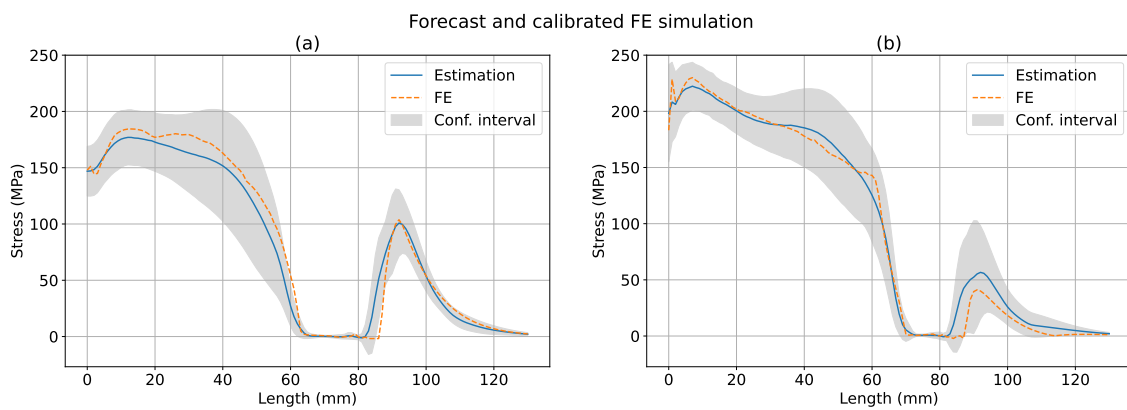


FIGURE 3.13: Forecasted maximum principal stress, FE simulation and confidence interval. (a) Camera side. (b) Torch side.

The Finite Elements simulation was used as a reference in Figure 3.12 and Figure 3.13, where we observe, generally, large confidence intervals on both sides of the specimen with respect to Figures 3.8 and 3.9, for which temperature data was observed. Overall, knowledge of the known and unknown parameters is enough to obtain predictions that represent the behaviour of the temperature and maximum principal stress fields but with a high degree of uncertainty, which can be greatly reduced by observation of the thermal images.

Forecasting alternatives

Section 3.4 explained how to use the estimation of the unknown parameters θ to forecast unseen results and in Figures 3.11, 3.12, and 3.13 we showed the predictions with their confidence intervals. Here, we provide some complementary forecasting results. In particular, we want to compare a forecast performed with and without observing the posterior theta values. Then, a comparison between the posterior covariance of a forecast and an estimation obtained by observing an infrared image located on the forecasted position. Only temperature results are shown. The comparison is restricted to the temperature field.

Figure 3.14 shows that temperature predictions obtained with and without observing the mean θ posterior. It can be observed that without observing the mean θ posterior the estimation is far from the infrared camera measurements, and forecasting is not possible. This indicates that the prior distribution of the unknown parameters might not be well-chosen or reflects a spectrum of values that is too large for this case. This justifies using the previously estimated unknown parameters to forecast states located further away, which result in a more accurate estimation of the temperature field.

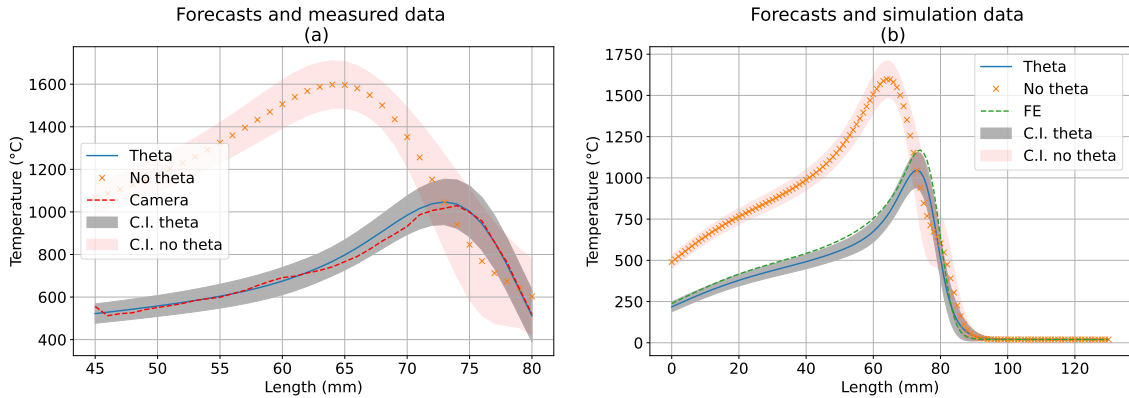


FIGURE 3.14: Forecasted temperature conditioning by the known parameters (μ) and by the known and unknown parameters (μ, θ) with their correspondent confidence interval. (a) Compared to experimental camera data. (b) Compared to FE results.

Additionally, we want to compare the posterior covariance of a forecasted estimation and an estimation obtained after observing an infrared image located on

the forecasted position, which can be seen in Figure 3.10. When observing the infrared image data, one can notice that the uncertainty on the temperature field, the uncertainty is significantly reduced in Figure 3.15.

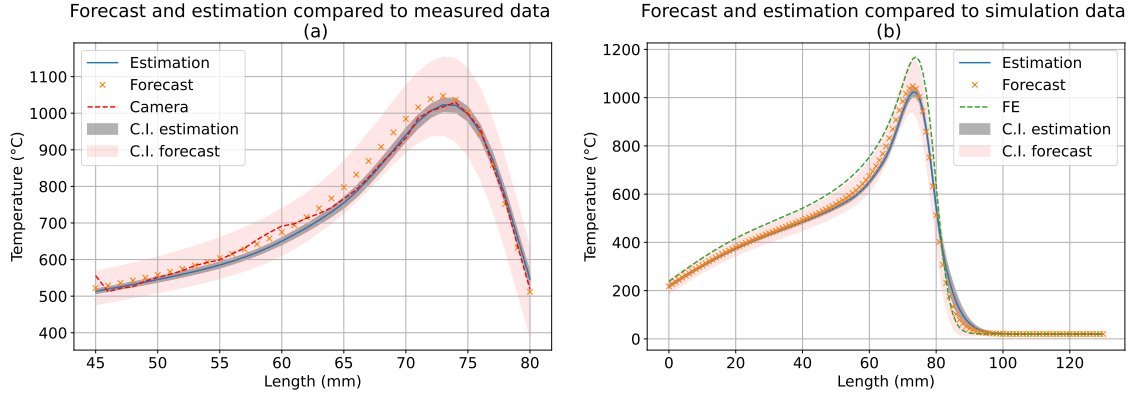


FIGURE 3.15: Forecasted temperature conditioning by the known and unknown parameters (μ, θ) and posterior estimation using the infrared camera data with their correspondent confidence interval. (a) Compared to experimental camera data. (b) Compared to FE results.

3.6 Conclusion

This chapter presented an offline/online approach for a multiphysics joint state and parameter estimation in real-time for a welding problem. This approach allows the estimation of observable and unobservable physical quantities and the calibration of parameters simultaneously.

In the offline stage, the local models are constructed from a numerical simulation database, discretised by the position of the heat source. We have shown that a simple SVD on different snapshot matrices is enough to construct the modes of each local model. Besides, the local multiphysics PPCAs allow to tackle several problems:

- Firstly, choosing a local model avoids using a global reduced basis. Thus, they capture local effects generated by the non-linear physics of the welding process.
- Secondly, the probabilistic framework considers the uncertainty in the observed data.
- Lastly, the multiphysical nature of the reduced bases allows the estimation of an unobserved quantity via the correlations with observed partial quantities such as surface temperature measurements.

The Gaussian assumptions that lead to the PPCAs have the advantage of an explicit solution for the MAP. Further algebraic manipulation helps attain the goal of a real-time application in the online stage.

The numerical tests, using synthetic and experimental data, show that the estimations are accurate compared to equivalent but costly Finite Elements simulations. The different data sources also showcase two situations: one where the heat source position is unknown and another where there is perfect knowledge of the torch position. A way to infer the heat source position that uses the simulation database is also given in this chapter. It relies on a surrogate model based on Gaussian Processes that links the highest temperature on one side of the specimen with the current torch position.

Despite the models being time-independent, we have shown how to use them to forecast future positions of the welding torch that have not been observed at the cost of more uncertainty in the predictions.

Chapter 4

Clustering local reduced bases

Abstract:

This chapter presents an alternative solution to the creation of the local PPCA models. In the previous chapter, each local model was created by selecting a position along the trajectory of the welding torch and assembling snapshot matrices for all known and unknown parameters (μ, θ) .

In this present chapter, a new automatic approach is explored in which a clustering algorithm determines how to separate the numerical data in order to create optimally accurate local models. Firstly, we will present three different ways of clustering the data, detailing the methods and showing examples. Secondly, a universal assignment function is constructed using a classification neural network. Such a function aims at assigning a given infrared temperature image to the most appropriate PPCA model in a physical sense. Three neural networks are designed, one for each clustering method. Finally, some results are shown using the new local bases.

Résumé :

Ce chapitre présente une méthode alternative pour la construction de modèles locaux d'ACP probabiliste. Dans le chapitre précédent, chaque modèle était créé en prenant une position dans la trajectoire de la torche de soudage et en assemblant une matrice avec les snapshots correspondant à tous les paramètres connus et inconnus (μ, θ) .

Maintenant, dans ce chapitre, une nouvelle approche automatique est explorée. Un algorithme de clustering déterminera comment séparer les données de simulations pour construire des modèles locaux optimaux. D'abord, on présentera trois métriques qui permettront de séparer les données, en précisant les méthodes et en montrant des exemples de leur application. Ensuite, on construira une fonction d'assignation universelle par un réseau de neurones de classification. Une telle fonction a pour but d'assigner le meilleur modèle d'ACP probabiliste à une image de température infrarouge. Trois réseaux de neurones seront entraînés (un par méthode) et ils seront utilisés pour réaliser des nouveaux tests numériques pour évaluer la performance des nouveaux modèles locaux.

Contents

4.1	Generalising the construction of local models	89
4.2	Clustering the simulation data	90
4.2.1	The K-medoids method	90
4.2.2	Description of the simulation data	92
4.2.3	Euclidean distance	92
4.2.4	Grassmann distance	93
4.2.5	Projection error in the state space	94
4.3	Offline stage results	96
4.3.1	Euclidean clustering	98
4.3.2	Grassmann clustering	99
4.3.3	Projection error clustering	101
4.3.4	Offline stage results recap	102
4.4	Online stage	103
4.4.1	Clustering labels	103
4.4.2	Optimal relabelling	105
4.4.3	Multi Layer Perceptrons	106
4.4.4	Training a classification Multi Layer Perceptron	109
4.5	Numerical results	110
4.5.1	Training of the MLP	111
4.5.2	3D temperature and stress estimation with noisy synthetic data	117
4.6	Conclusion	128

4.1 Generalising the construction of local models

The previous chapter presented an offline-online approach for the real-time estimation of partially observed and non-observed physical fields in a welding application. This method relies on the construction of local reduced bases with a Gaussian structure that are consequently conditioned by observations in the online stage. A critical point that was not explored was how to partition the data to create the local models.

In most situations, time and space might be a natural way of separating the local models. This is the case of the PVR experiment, where the natural choice is to use the position of the heat source as the partition. There are numerous examples in the literature that use time for partitioning, for example in Dihlmann et al. [44] or Drohmann et al. [46]. In the case of parametrised problems, other studies propose a partition of the parameter space [62] or a partition on the state space [168, 8, 131]. More recently, novel approaches such as ROM-Nets [41] have appeared, making use of recent developments in neural networks to solve classification problems related to data partition and local reduced bases.

Most of these approaches use some notion of distance or dissimilarity measure in order to partition the data. This partitioning of the data in the offline stage can be achieved by clustering. Clustering methods allow separating the data in groups called clusters such that all elements in a cluster are similar to each other according to some metric and, at the same time, are distinct from elements in other clusters [55].

The idea is to run a clustering algorithm that will separate all the simulation snapshots into K clusters C_i , $\forall i \in \{1, \dots, K\}$ according to a particular criterion. A reduced basis Φ^{C_i} will be computed from the snapshots in C_i . The construction of the local PPCA models is the same independently of the clustering method.

Let us assume that all the snapshots from the parametrical study are divided into K clusters. Following the methodology presented in the previous chapter, a snapshot matrix \mathbf{S}_i is defined per cluster C_i as:

$$\mathbf{S}_i = \{\mathbf{s}_j, j \in \{1, \dots, N_S\} : \mathbf{s}_j \in C_i\}, \quad \forall i \in \{1, \dots, K\}, \quad (4.1)$$

where N_S is the total number of snapshots, and \mathbf{s}_j identifies a multiphysics joint state vector, such as they are presented in the previous chapter. Let us remind that a joint state vector contains a temperature field, a principal stress field, and the known and unknown parameters:

$$\mathbf{s}_j = \begin{pmatrix} \mathbf{T}^j \\ \boldsymbol{\sigma}_I^j \\ \mu^j \\ \theta^j \end{pmatrix} \in \mathbb{R}^{2 \times N_n + N_\mu + N_\theta} \quad (4.2)$$

where N_n is the number of nodes in the mesh, N_μ is the number of known parameters and N_θ is the number of unknown parameters, following the notations already introduced in Chapter 3.

Once the snapshot matrices are assembled, the local PPCA models are computed by SVD and the Maximum Likelihood estimators (see Chapter 3 for more details). The offline-online approach is summed up as follows:

Offline-online approach

- Offline stage:
 1. Partition the data into K clusters C_i , $\forall i \in \{1, \dots, K\}$.
 2. Compute a SVD on a snapshot matrix formed by all snapshots in C_i $\forall i \in \{1, \dots, K\}$.
 3. Select the amount of modes n_{ϕ_i} in the SVD and compute the PPCA model.
- Online stage:
 1. Receive a new infrared image.
 2. Identify the most adequate local PPCA model.
 3. Compute the MAP of the posterior distribution of the multiphysics vector conditioned by the infrared image and its confidence interval.

The following sections will focus on partitioning the data and selecting the adequate local model for new input data.

4.2 Clustering the simulation data

The previous section introduced some notions on partitioning the data in a simulation database in order to create local reduced-order models. Data clustering depends on a set of points in a certain space \mathcal{X} and a metric δ :

$$\begin{aligned} \delta : \mathcal{X} \times \mathcal{X} &\longrightarrow \mathbb{R}^+ \\ (x, y) &\longrightarrow \delta(x, y) \end{aligned}$$

that allows determining how similar or dissimilar two data points are. The choice of an adequate metric is of utmost importance.

The following subsections will explain the clustering method that was selected in this work and describe the data points. Finally, three metrics are presented. These three metrics are the Euclidean distance in the solution space, the projection error between snapshots and the Grassmann distance.

4.2.1 The K-medoids method

The K-medoids method [117, 148] is going to be used in the following sections to cluster the simulation data. This method has been chosen in the present work as it requires very few input parameters: the number of clusters and a dissimilarity matrix. Using a dissimilarity matrix allows to try a large variety of metrics because the algorithm is not based on the properties of a particular metric. A dissimilarity matrix is a symmetrical matrix D where each element $D_{ij} = \delta(x_i, x_j)$ is a pseudo distance between data points x_i and x_j . To be numerically efficient, only half of

the matrix is computed. One more advantage of the K-medoids algorithm is that each cluster's medoid is a dataset point, unlike similar methods like the K-means algorithm. Medoids m_k are defined as the points of each cluster that minimize the average dissimilarity to all other elements in their cluster:

$$m_k = \arg \min_{x_i \in C_k} \frac{1}{\#C_k} \sum_{x_j \in C_k} \delta(x_i, x_j) \quad (4.3)$$

Algorithm 4.1: K-medoids algorithm

Input: Dissimilarity matrix $D \in \mathbb{R}^{p \times p}$, number of clusters K
Output: Labels $\{l_i\}_{i=1}^n$ where $l_i \in \llbracket 1, K \rrbracket$, medoids $M = \{m_i\}_{i=1}^K$
Initialisation:
 Select the initial medoids $M = \{m_1, \dots, m_K\}$:
 1 - Calculate v_j , $j = 1, \dots, p$:

$$v_j = \sum_{i=1}^p D_{ij}, \quad j = 1, \dots, p$$

 2 - Sort $\{v_j\}_{j=1}^p$ in ascending order $\text{sorted}(\{v_j\}_{j=1}^p)$ and assign the first K indices j as medoids $\{m_1, \dots, m_K\}$
 3 - Assign a label l_i to each data point:

$$l_i = \arg \min_{k=1, \dots, K} D_{i, m_k}$$

 4 - Create the clusters C_1, \dots, C_K :

$$C_k = \{i \in \llbracket 1, p \rrbracket : l_i = k\} \quad k = 1, \dots, K$$

 5 - Create an empty list of medoids M^{old} // This is used as the stopping criterium
while $M \neq M^{old}$ **do**
 Store the medoids of the previous iteration in M^{old} : $M^{old} \leftarrow M$
 Update the medoids:
 foreach Cluster C_k **do**
 Compute the sum of dissimilarities of each element of the cluster to the others and select the one that minimizes it as new medoid m_k^* :

$$v_i = \sum_{j \in C_k} D_{ij}, \quad \forall i \in C_k$$

$$m_k^* = \arg \min_{i \in C_k} v_i$$

 end
 Store the new medoids in M :

$$M \leftarrow \{m_1^*, \dots, m_K^*\}$$

 Assign new labels and create new clusters:

$$l_i = \arg \min_{k=1, \dots, K} D_{i, m_k^*} \quad C_k = \{i \in \llbracket 1, p \rrbracket : l_i = k\} \quad k = 1, \dots, K$$

end

There are several algorithms of the K-Medoids method, of which the most popular ones are the PAM, CLARA and CLARANS algorithms[82, 147]. Our in-house implementation is based on the algorithm proposed by Park et al. [117] and is shown in 4.1. This algorithm is divided into an initialization step and an update step. The initialization step selects a first set of candidate medoids and assigns each element to

a medoid. The update step is a loop that runs until two consecutive iterations yield the same result. In each iteration of the loop, we find the element of each cluster k that minimizes the sum of the dissimilarities to other elements in its cluster. Then, each data point is reassigned to the medoid that minimizes the dissimilarity.

4.2.2 Description of the simulation data

The starting point of this section is the simulation database resulting from the parametrical study presented in Chapter 2. Let us recall that the parametrical study is composed of $M = 128$ simulations of PVR experiments depending on two sets of parameters: $\mu = (v, Q)$ the known operational parameters and $\theta = (a_f, b, c, \eta, K)$ the unknown parameters of the equivalent heat source model. Experts delimit the values of each parameter, and the parametrical space is explored using Latin Hypercube Sampling. Each simulation is run for $N_t = 260$ time steps that correspond to 260 positions of the welding torch in a regular assimilation grid. In total, there are $N_S = N_t \times M = 33280$ snapshots that are entirely determined by the triplet (μ, θ, P) , corresponding to the known parameters (speed and power), the unknown parameters and the position of the heat source. The snapshots are joint state vectors that contain a nodal temperature field, mechanical fields and the known and unknown parameters. This was presented in Equation (3.7) for positional state vectors. Here we will index each snapshot by an integer $j \in \{1, \dots, N_S\} = \mathcal{J}$:

$$\mathbf{s}_j = \mathbf{s}(\mu_i, \theta_i, P^k) = \begin{pmatrix} \mathbf{T}^j \\ \boldsymbol{\sigma}_I^j \\ \mu^j \\ \theta^j \end{pmatrix} \quad (4.4)$$

where $i \in \{1, \dots, M\}$ indexes the parameters in the DOE and $k \in \{1, \dots, N_t\}$ indexes the positions. The index j is calculated from :

$$j = (i - 1)N_t + k \quad (4.5)$$

4.2.3 Euclidean distance

One of the most commonly used, if not the most used, distance in clustering and machine learning algorithms is the Euclidean distance. The Euclidean distance, noted δ_E , between two n -dimensional vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ reads:

$$\delta_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4.6)$$

The input data for this clustering method are the snapshots \mathbf{s}_j in the state space. The temperature, stress and parameters are all compared at the same time. The main advantage of the Euclidean distance is its simplicity and the numerous optimised implementations to compute it for large vectors. This is helpful for the construction of the dissimilarity matrix.

There is, however, a disadvantage to this metric in the context of Model Order Reduction. The Euclidean distance between two snapshots u, v that are only different by a multiplicative factor might be very high. The clustering method will assign them to different clusters, but it is clear that they can be expressed by the same basis function. The other metrics should deal with this disadvantage.

In order to cluster the data, we could have considered using the triplet $\mathbf{x}_j = (\mu_i, \theta_i, P^k)$ as input data. This version of the Euclidean clustering greatly increases the computation time because the involved vectors in the PVR application case are 8-dimensional ($N_\mu = 2$ and $N_\theta = 5$). Each multiphysics snapshot s_j is represented by a vector $\mathbf{x}_j = (\mu_i, \theta_i, P^k)$ that contains the parameters used for its generation and the current torch position, where i, j and k are related by Equation (4.5).

The different parameters and fields are not expressed in the same scale. For example, in the PVR application case, the power of the welding torch is expressed in the hundreds of watts whilst the efficiency in the Goldak equivalent heat source model is a number between 0 and 1. Before computing the distance matrix, all data is rescaled between 0 and 1 according to the minimum and maximum values observed, with 0 assigned to the minimum (\mathbf{x}_{min}) and 1 to the maximum (\mathbf{x}_{max}):

$$\bar{\mathbf{x}}_j = \frac{\mathbf{x}_j - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} \quad (4.7)$$

Clustering the snapshots by the input parameters ignores that different parameter values might lead to the same solution. If the Euclidean distance between two sets of parameters that yield the same solution is sufficiently large, the snapshots will be separated in different clusters. This is contrary to the clustering objective and, thus, this method will not be investigated any further.

4.2.4 Grassmann distance

The second proposed metric is the Grassmann distance, which has already been used as a dissimilarity metric in Reduced Order Modelling (ROM) applications. In particular, the Grassmann manifold and its geodesic distance have been used by Amsallem and Farhat [5], or by Mosquera [112] to interpolate local bases.

The Grassmann distance, already introduced in the state of the art, is a geodesic distance on the a Grassmann manifold [175]. The Grassmann manifold $\mathcal{G}(m, n)$, with $m \leq n$, is defined as the set of all m -dimensional subspaces of \mathbb{R}^n . Two reduced bases $\Phi \in \mathbb{R}^{\mathcal{N} \times n_\phi}$ and $\Psi \in \mathbb{R}^{\mathcal{N} \times n_\phi}$ span subspaces $\text{span}(\Phi), \text{span}(\Psi) \subset \mathbb{R}^{\mathcal{N}}$. Those two subspaces are on the Grassmann manifold $\mathcal{G}(n_\phi, \mathcal{N})$. Thus, a way of comparing two reduced bases is to compare their respective spanned subspaces on a Grassmann manifold. The Grassmann distance between Φ and Ψ , $\delta_G(\Phi, \Psi)$, is defined as:

$$\delta_G(\Phi, \Psi) = \sqrt{\sum_{n=1}^{n_\phi} \lambda_n^2} \quad (4.8)$$

where λ_n , $n = 1, \dots, n_\phi$ are the n^{th} principal angles between Φ and Ψ , which are computed from a SVD of $\Psi^T \Phi = V \Sigma W^T$:

$$\lambda_n = \cos^{-1}(\sigma_n) \quad (4.9)$$

with σ_n the n^{th} singular value. This distance is the geodesic distance on $\mathcal{G}(n_\phi, \mathcal{N})$ when considering the manifold as a Riemannian manifold [175]. For more in-depth details on the Grassmann manifold and its properties, we refer to Amsallem's or Mosquera's work [5, 112].

The dissimilarity metric is defined between subspaces, but our dataset is formed by snapshots. In order to use this metric, we need existing local bases. The proposed solution is to run a preliminary Euclidean clustering with a large number of clusters K . In this case, we will choose $K = 4000$. This produces a new dataset composed of 4000 preliminary local reduced bases from which a distance matrix is computed. Other methods were proposed for this partitioning: overlapping time intervals and balls of a specific user-defined radius in the parameter space. The entire procedure is summed up as follows:

Grassmann clustering procedure

1. Partition the data into $K = 4000$ clusters with the Kmedoids algorithm using Euclidean distance.
2. Construct preliminary POD bases via SVD on the clustered snapshots.
3. Compute the Grassmann distance matrix between the K preliminary bases.
4. Partition the data into $K = 260$ clusters with the Kmedoids algorithm and the Grassmann distance matrix.
5. Construct the POD bases via SVD on the clustered snapshots.
6. Construct the local PPCA model per cluster from the POD bases.

The initial number of clusters $K = 4000$ determines the computational complexity of the distance matrix calculation. It was already indicated that for the sake of numerical efficiency, only the upper diagonal is computed. This is, however, still a considerable number of distances between high dimensional bases, in particular, $\frac{K(K-1)}{2}$. For $K = 4000$, the total number of distances to compute is almost 8×10^6 , which takes between two and three days of parallel computation on a scientific cluster.

4.2.5 Projection error in the state space

The final metric that is considered is the projection error between snapshots. This idea, proposed by Amsallem and Haasdonk [6], aims at partitioning the data according to the properties of the solution in the state space. We define the projection error between two vectors \mathbf{u}, \mathbf{v} in the state space :

$$\delta_P(\mathbf{u}, \mathbf{v}) = \left\| \mathbf{u} - \frac{\mathbf{v}\mathbf{v}^T \mathbf{u}}{\mathbf{v}^T \mathbf{v}} \right\|_2 \quad (4.10)$$

An advantage over the Euclidean distance for MOR applications is that it captures effects of scale invariance, as indicated by Amsallem. Indeed, the dissimilarity

between two vectors \mathbf{u}, \mathbf{v} and $\mathbf{u}, \alpha\mathbf{v}$, with α a non-zero real number, is the same. The vectors \mathbf{v} and $\alpha\mathbf{v}$ can be expressed by the same basis function and should be clustered together. This property is easily demonstrated:

Proposition 1. *The metric δ_P is scale invariant:*

$$\delta_P(\mathbf{u}, \alpha\mathbf{v}) = \delta_P(\mathbf{u}, \mathbf{v}), \quad \forall \alpha \in \mathbb{R}^*$$

Proof. Let α be a non-zero real number.

$$\delta_P(\mathbf{u}, \alpha\mathbf{v}) = \left\| \mathbf{u} - \frac{(\alpha\mathbf{v})(\alpha\mathbf{v})^T \mathbf{u}}{(\alpha\mathbf{v})^T (\alpha\mathbf{v})} \right\|_2 = \left\| \mathbf{u} - \frac{\alpha^2 \mathbf{v} \mathbf{v}^T \mathbf{u}}{\alpha^2 \mathbf{v}^T \mathbf{v}} \right\|_2 = \delta_P(\mathbf{u}, \mathbf{v})$$

□

It follows that the projection error between a vector and a scaled version of itself is nil, assuring that scaled versions of the same solution are clustered together.

Corollary 1.1.

$$\delta_P(\mathbf{u}, \alpha\mathbf{u}) = 0, \quad \forall \alpha \in \mathbb{R}^*$$

Proof. Let α be a non-zero real number. According to the previous proposition $\delta_P(\mathbf{u}, \alpha\mathbf{u}) = \delta_P(\mathbf{u}, \mathbf{u})$.

$$\delta_P(\mathbf{u}, \alpha\mathbf{u}) = \delta_P(\mathbf{u}, \mathbf{u}) = \left\| \mathbf{u} - \frac{\mathbf{u}(\mathbf{u}^T \mathbf{u})}{\mathbf{u}^T \mathbf{u}} \right\|_2 = \left\| \mathbf{u} - \mathbf{u} \right\|_2 = 0$$

□

Alternative hierarchical clustering algorithm

Computing a dissimilarity matrix for this metric is extremely expensive. The total number of operations is $\frac{N_S(N_S-1)}{2}$, which in this case amounts to 553762560 evaluations of the projection error between vectors of dimension $2 \times N_n + N_\mu + N_\theta = 208209$. Instead, we will follow the hierarchical algorithm presented by Amsellem [6]. The hierarchical algorithm does not take the number of clusters as input, instead there is a criterion $\epsilon_P > 0$ that ensures that the projection error between all the elements in the same cluster is sufficiently small:

$$\forall i, j \in \mathcal{J} \text{ such that } \mathbf{s}_i, \mathbf{s}_j \in C_k \implies \delta_P(\mathbf{s}_i, \mathbf{s}_j) < \epsilon_P \quad (4.11)$$

The proposed algorithm is a divisive hierarchical algorithm, meaning that at first, all the data are part of a single cluster and iteratively, the datapoints are split into smaller clusters. The hierarchical algorithm is initialised by selecting an anchor point and calculating the projection error of all snapshots with respect to the anchor. In successive iterations, the snapshot that realises the highest error is selected as a new anchor point and its cluster is divided into two clusters. All snapshots whose projection error is smaller with respect to the new anchor point are selected as part of the new cluster. The procedure is repeated until the maximum error is smaller than the stop criterion ϵ_P . Although the algorithm presented in 4.2 stem from [6], its initialization step has been modified in the present work. Indeed, we have observed that in our dataset, starting with the snapshot with the highest L^2 norm

leads to a very large first cluster that doesn't concentrate data in any meaningful way. Instead of choosing the snapshot with the highest L^2 norm, the median value has been chosen. More details on this are given in Annexe A.

Algorithm 4.2: Projection Error Clustering

Input: Snapshots s_i , ϵ_P

Output: Labels $\{l_i\}_{i=1}^n$ where $l_i \in \llbracket 1, \dots, K \rrbracket$, anchor points $\{m_i\}_{i=1}^K$

Initialisation:

Select the first anchor point m_1 and compute the maximum error to the anchor ϵ_1 :

$m_1 = \text{median}(\|s_j\|_{L^2}), \quad \forall j \in \mathcal{J}$

$C_1, \epsilon_i = \max_s \delta_P(s, m_1)$ // initialise first cluster

$k = 1$ // counts the current amount of clusters

while $\exists i \in \{1, \dots, k\}$ with $\epsilon_i \geq \epsilon_P$ **do**

 Find which cluster contains the highest error:

$i = \arg \max_{i \in \{1, \dots, k\}} \epsilon_i$

 Search new anchor point:

$k \leftarrow k + 1$

$m_k = \arg \max_{s \in C_i} \delta_P(s, m_i)$

 Create new clusters $C_k = \{s \in C_i : \delta_P(s, m_i) > \delta_P(s, m_k)\}$ and

$C_i \leftarrow C_i \setminus C_k$

 Update M:

$M = \{m_1, \dots, m_k\}$

 Compute new errors $\epsilon_i, \forall i \in \{1, \dots, k\}$:

$\epsilon_i = \max_{s \in C_i} \delta_P(s, m_i)$

end

A problem that can arise when clustering the snapshot data is the appearance of too small clusters, including unitary clusters. This is not desirable, as the resulting local models would not be representative of many states. A second step to the algorithm has been added with respect to Amsallem's. If a cluster has less elements than a certain threshold t_{size} , we will find the closest cluster in terms of projection error between representatives and merge them in order to obtain bigger clusters. The following Algorithm 4.3 adds this step. It should be noted that there is a new parameter t_{size} that sets the minimum size of the clusters.

Using this algorithm instead of calculating the dissimilarity matrix saves a lot of time. The first two iterations need $N_S = 33280$ evaluations of the projection error, but the following iterations keep decreasing the amount of evaluations needed.

4.3 Offline stage results

This section compares the results obtained using each different dissimilarity metric. The number of clusters K is selected to match the number of local bases in the previous chapter: 260. For the projection error clustering, the stopping criteria is set to $\epsilon_i = 0.245$, which yields 255 clusters.

Algorithm 4.3: Modified Projection Error Clustering

Input: Snapshots s_i , ϵ_P , threshold t_{size}
Output: Labels $\{l_i\}_{i=1}^n$ where $l_i \in \llbracket 1, \dots, K \rrbracket$, anchor points $\{m_i\}_{i=1}^K$
Initialisation:
 Select the first anchor point m_1 and compute the maximum error to the anchor ϵ_1 :
 $u_1 = \text{median}(\|s_j\|_{L^2}), \quad \forall j \in \mathcal{J}$
 $C_1, \epsilon_1 = \max_s \delta_P(s, m_1)$ // initialise first cluster
 $k = 1$ // counts the current amount of clusters
while $\exists i \in \{1, \dots, k\}$ with $\epsilon_i \geq \epsilon_P$ **do**
 Find which cluster contains the highest error:
 $i = \arg \max_{i \in \{1, \dots, k\}} \epsilon_i$
 Search new anchor point:
 $k \leftarrow k + 1$
 $m_k = \arg \max_{s \in C_i} \delta_P(s, m_i)$
 Create new clusters $C_k = \{s \in C_i : \delta_p(s, m_i) > \delta_p(s, m_k)\}$ and
 $C_i \leftarrow C_i \setminus C_k$
 Update M:
 $M = \{m_1, \dots, m_k\}$
 Compute new errors $\epsilon_i, \forall i \in \{1, \dots, k\}$:
 $\epsilon_i = \max_{s \in C_i} \delta_P(s, m_i)$
 foreach $C_i \quad i \in \{1, \dots, K\}$ **do**
 if $\#C_i < t_{size}$ **then**
 Find the cluster whose anchor is closest to m_i :
 $\arg \min_{k \in \{1, \dots, K\}} \delta_P(m_i, m_k)$
 Redefine C_k, C_i and M :
 $C_k \leftarrow C_k \cup C_i, \quad C_i = \emptyset, \quad M \leftarrow M \setminus \{m_i\}$
 end
 end
 Reset indices and redefine $K \leftarrow \#M$
end

Visualisation of the clustering results is challenging due to the high dimensions of the data and the number of clusters. In order to analyse the clusters obtained with each metric, one cluster is arbitrarily selected. The observed trends issued from this analysis need to be validated by some notion of representativity. In this case, we have decided to measure the intra-cluster variance of the interquartile range of the scaled parameters v (speed), Q (power) and P (position). The interquartile range IQR is defined as the difference between the 75th (P_{75}) and the 25th (P_{25}) percentiles:

$$IQR = P_{75} - P_{25} \quad (4.12)$$

where a percentile P_n is the point below which the $n\%$ of the data is found.

This means that for a given cluster C_k , we will define a variable IQR^k that measures the spread of the snapshots by observing the distribution of the parameters that determine them. The mean of this variable indicate that the parameters in each

cluster are spread in a similar way. Thus, if its variance is close to zero, the trends that we can observe in one cluster are transferable to other clusters.

Figure 4.1 shows a boxplot of the cluster sizes produced by each metric. This type of plot, introduced by Tukey [161], summarizes the data with a box that extends from the first to the third quartiles (Q_1 and Q_3 , respectively). The median is depicted by an orange line. The upper and lower boundaries of the whiskers (u_b and l_b , respectively) are calculated by:

$$u_b = Q_3 + 1.5 \times IQR \quad (4.13a)$$

$$l_b = Q_1 - 1.5 \times IQR \quad (4.13b)$$

All data beyond the whiskers' boundaries are represented individually.

The first noticeable difference between the three clustering methods is that only the Euclidean distance produces homogeneous clusters. Indeed, the Projection error and Grassmann clusterings produce heterogeneous clusters with some extremely large clusters compared to the rest.

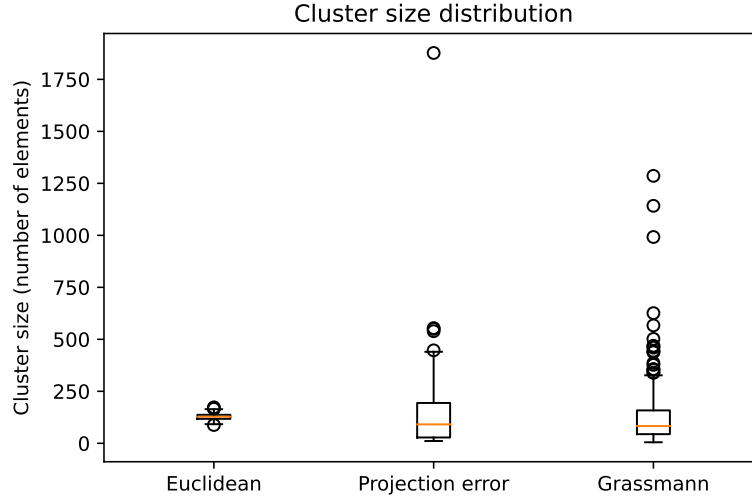


FIGURE 4.1: Distribution of cluster sizes for each dissimilarity metric.

4.3.1 Euclidean clustering

The Euclidean clustering is done in a straightforward manner. The distance matrix is computed from the snapshot list in parallel. Then, the K-medoids algorithm separates the data into 260 clusters. Figure 4.2 shows a boxplot of the distribution of the scaled speed, power and position of the snapshots selected for cluster number 3, which was selected arbitrarily.

One can notice that the Euclidean distance concentrates snapshots with similar power and position. This seems like a natural consequence of how power and position affect the heat on the specimen. Power affects the temperature values, and the position changes the form of the heat distribution on the specimen completely.

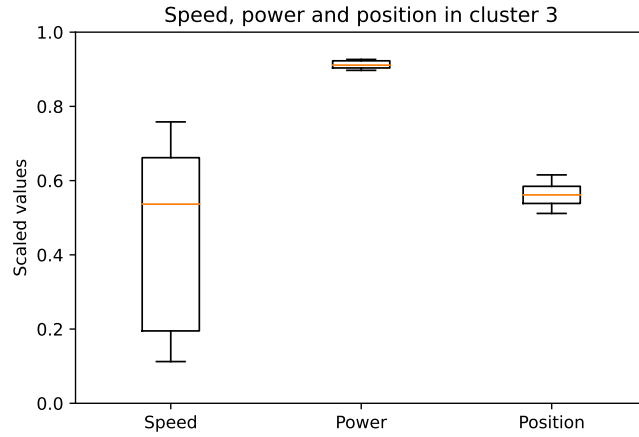


FIGURE 4.2: Distribution of speed, power and position for an arbitrary cluster (Euclidean distance).

Considerable variations in these parameters might create vastly different fields from the perspective of the Euclidean distance.

The interquartile range's mean and variance of each of the scaled parameters are given in Table 4.1. The IQR's mean and variance for position and power suggest that all the clusters concentrate the snapshot according to the value of power and position, which validates the previous comments. However, there is no such behaviour regarding the speed. The range of speed values is greater in most clusters.

TABLE 4.1: Interquartile range mean and variance of each scaled parameter (Euclidean clustering).

	IQR(Speed)	IQR(Power)	IQR(Position)
Mean	0.420207	0.021012	0.042119
Variance	0.040117	0.000044	0.000016

4.3.2 Grassmann clustering

The Grassmann clustering is performed in two consecutive steps. It has already been noted that the Grassmann distance is not defined between snapshots, as they are not points in the Grassmann manifold. There is a need for a first partitioning of the data in order to create preliminary reduced bases to be compared with the Grassmann distance. It should be pointed out that this first partitioning might introduce a bias in the way the Grassmann clusters are constructed. The first partitioning consists of a Euclidean clustering performed on the parameter space for a very large number of clusters K , in this case, $K = 4000$.

According to Figure 4.1, the Grassmann cluster produces clusters of varying sizes instead of the very homogeneous clusters obtained with the Euclidean distance. Figure 4.3 displays cluster 10, which is the largest cluster obtained with the Grassmann

distance. Unlike the Euclidean clustering, only snapshots with similar positions are clustered together, with significant variance in speed and power. The interquartile range variance, presented in Table 4.2, shows, in agreement with the cluster in Figure 4.3, that most clusters tend to group snapshots around the same position, with the speed and power having greater variety.

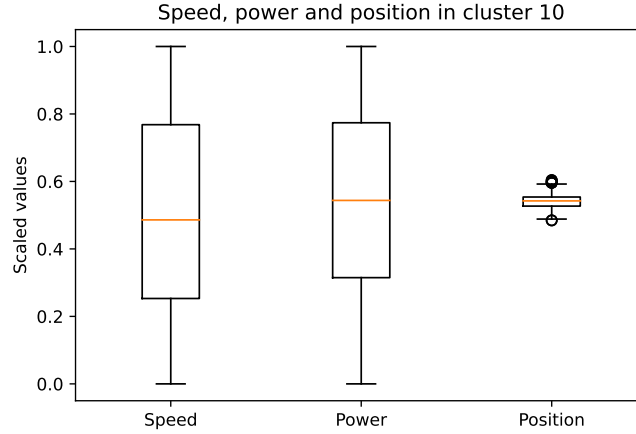


FIGURE 4.3: Distribution of speed, power and position for an arbitrary cluster (Grassmann distance).

TABLE 4.2: Interquartile range mean and variance of each scaled parameter (Grassmann clustering).

	IQR(Speed)	IQR(Power)	IQR(Position)
Mean	0.344514	0.380694	0.016801
Variance	0.022942	0.029629	0.000033

The Grassmann distance creates some very big clusters. We are interested in whether these clusters capture a stationary region in the specimen that is very easily reproduced by the reduced bases. The welding torch positions that are represented in the six largest clusters were analysed and summed up in a histogram in Figure 4.4. These six clusters are those with more than 500 snapshots in them (1286, 1142, 992, 626, 567 and 502, respectively), which accounts for 15.31% of the snapshots. In agreement with Table 4.2, these clusters gather around separated positions.

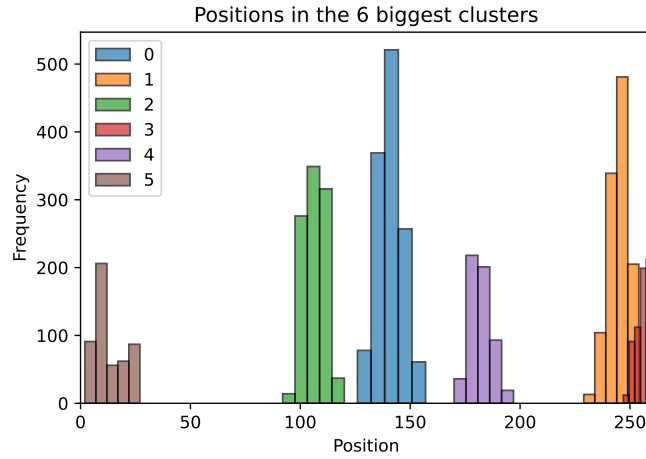


FIGURE 4.4: Welding torch positions represented in the 6 largest clusters (Grassmann clustering).

4.3.3 Projection error clustering

Finally, the last clustering method involves the projection error between snapshots in the solution space. One should be reminded that the clustering method is not K-medoids, as in the previous examples, but the algorithm presented in Algorithm 4.3. The stopping criterion is set to $\epsilon_P = 0.245$ and the minimum cluster size threshold is set to 10. This configuration yields a total of 255, close to the 260 of the previous methods.

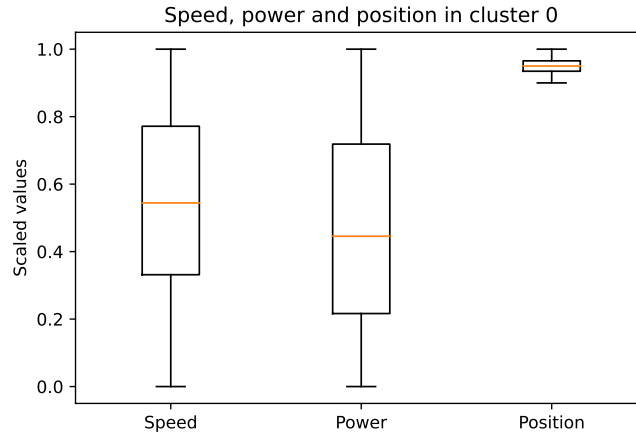


FIGURE 4.5: Distribution of speed, power and position for the largest cluster (Projection error).

Once again, we have decided to show the results on the largest cluster produced by this method in Figure 4.5. In this cluster, the snapshots are concentrated by the position, with variety in the speed and power values. The IQR in Table 4.3

TABLE 4.3: Interquartile range mean and variance of each scaled parameter (Projection error clustering).

	IQR(Speed)	IQR(Power)	IQR(Position)
Mean	0.303896	0.415119	0.010011
Variance	0.009896	0.010675	0.000019

show that this behaviour is shared by other clusters, where the spread of snapshots according to the three parameters is similar.

The largest cluster contains 1877 snapshots with the torch positioned at the end of the specimen. This suggests that the temperature snapshots when the torch is arriving at the final position are very similar in the sense of vector spaces even for signification variations of speed and power. Indeed, the 5 largest clusters are concentrated around the final position, see Figure 4.6. These five clusters account for 11.87 % of the snapshots and have, respectively, 1877, 554, 550, 539 and 447 snapshots.

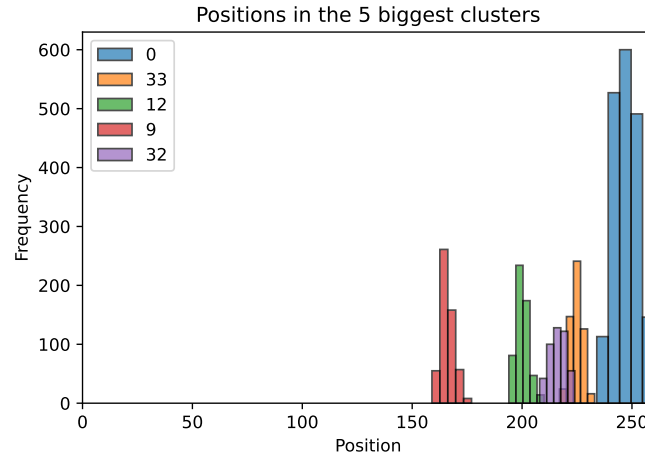


FIGURE 4.6: Welding torch positions represented in the 5 largest clusters (Projection error clustering).

4.3.4 Offline stage results recap

The previous sections have shown the results of clustering the simulation data by applying three different dissimilarity metrics. The most remarkable result is that all three metrics tend to group data according to the position of the heat source in the selected snapshots. This supports the physical intuition that led to the separation of local models by position in Chapter 3.

The Euclidean distance also grouped snapshots according to the power of the snapshots, indicating a sensibility in the Euclidean distance to variations in power that lead to temperature fields with an extensive range of values. This phenomenon

is mitigated for the other two metrics. Grassmann clustering groups snapshots according to the similarity between the spanned subspaces of reduced bases calculated from the snapshots. Intuitively, two similar fields with a difference in scale will span similar subspaces because they can be represented by the same basis function with a different coefficient. Projection Error clustering has been proven to be scale-invariant.

The other major difference is in the size of the clusters. Whilst the Euclidean clustering produces homogeneous clusters, the other two methods were observed to create very heterogeneous clusters in terms of size when sampling the parameter space by using a Latin Hyper Cube algorithm.

Finally, the results were analysed according to the known parameters $\mu = (v, Q)$ and the position of the torch. Additional analysis on the unknown parameters θ was made, but it is not shown in this document, as no relevant pattern was observed and including them in the boxplots make them more difficult to read.

4.4 Online stage

The online stage is divided into two distinct steps: a classification step in which new input data, in this case a partial observation of the temperature field, is assigned the most adequate local surrogate model, and an estimation step in which the local model produces a 3D estimation of the entire temperature and maximum principal stress fields, as well as an estimation of the unknown parameters. In this section, we will focus on the classification step, whilst we refer to Chapter 3 for details on how the estimations are computed.

In the online stage, an infrared image has to be assigned to the adequate local surrogate model, which is associated with a cluster. The classification step is, thus, reduced to finding a label for a vector containing the partially observed temperature and known parameters, which points to a local surrogate model. The three clustering methods introduced in the previous section produce a natural classification tool to identify the best cluster for the input data. However, this assignment could be computationally expensive for large numbers of clusters K .

In this work, we aim at finding a universal classification function that assigns the most appropriate PPCA model inexpensively and independently of the clustering method used to construct them. This universal classification will be a Multi Layer Perceptron (MLP) neural network. In order to train the MLP, we will need a set of labels for all of the simulation snapshots. The following subsections will introduce the labelling methods that naturally arise from the clustering methods and a new optimal labelling method that takes into account the reconstruction error of the snapshot with the reduced bases computed for each cluster. Finally, the chosen MLP model is explained in detail.

4.4.1 Clustering labels

As it was indicated at the beginning of this section, the clustering methods provide a natural classification tool by identifying each snapshot with the cluster that it was assigned into. The assignment of a label to new data is done differently for each clustering method and will rely on the use of the clustering metric, which could prove

computationally expensive. Moreover, the training data is composed of multiphysics state vectors whilst in the online stage the data will consist exclusively of partial surface measurements of temperature.

In the online stage, the input data consists only of partial superficial temperature data and the known parameters μ : speed and power. This partial knowledge can be represented by a boolean operator H , which has already been introduced in Chapter 3 that restrains the current knowledge of the multiphysics vector to the available data:

$$\mathbf{T}_j = H\mathbf{s}_j \in \mathbb{R}^{N_d} \quad (4.14)$$

where T_j is a vector containing the observed temperature field and the known parameters of the snapshot s_j , and N_d is the dimension of the observed data vector, corresponding to the number of nodes observed with the camera and the number of known parameters.

Euclidean clustering

The Euclidean clustering assignment of a label to a new observed vector is straightforward. The cluster for a certain snapshot s_j can be calculated as the cluster whose centroid realises the minimum distance to s_j . The label l_j assigned to a multiphysics snapshot s_j is:

$$l_j = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_E(\mathbf{s}_j, \mathbf{m}_k) \quad (4.15)$$

where \mathbf{m}_k is the medoid associated to the cluster C_k . The labelling method in Equation (4.15) still holds for the observed snapshots \mathbf{T}_j :

$$l_j = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_E(\mathbf{T}_j, H\mathbf{m}_k) \quad (4.16)$$

and can be used for new unknown input infrared images.

Projection Error clustering

The same idea presented for the Euclidean clustering could be applied to the Projection Error clustering. The label is assigned by computing the projection error to the representatives \mathbf{m}_k of each cluster. However, the computation time increases significantly for large values of K . Here, $K = 255$ makes the computation quite expensive, unlike for the Euclidean case. The label assignment is:

$$l_j = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_p(\mathbf{s}_j, \mathbf{m}^k) \quad (4.17)$$

where \mathbf{m}^k is the representative of cluster C_k , as presented in Algorithm 4.3.

Furthermore, an alternative version could also compute the projection error with respect to the reduced bases Φ^k of each cluster C^k , offering a more meaningful labelling in the context of MOR:

$$l_j = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_p(\mathbf{s}_j, \Phi^k) = \arg \min_{k \in \llbracket 1, K \rrbracket} \|\mathbf{s}_j - \Phi^k(\Phi^k)^T \mathbf{s}_j\|_2 \quad (4.18)$$

In this case, the projection error is related to the projection of \mathbf{s}_j onto the reduced basis Φ^k .

As for the Euclidean assignment, both of these labelling methods hold when considering partially observed snapshots \mathbf{T}_j by computing the Projection error between \mathbf{T}_j and $H\mathbf{m}_k$ or $H\Phi^k$, respectively. The labelling in these cases are:

$$l_j = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_p(\mathbf{T}_j, H\mathbf{m}^k) \quad (4.19)$$

and:

$$l_j = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_p(\mathbf{T}_j, H\Phi^k) \quad (4.20)$$

Grassmann clustering

The assignment with the Grassmann clustering cannot be done in the same way as in Equations (4.16) or (4.19). In both cases, the representatives of each cluster, are multiphysics snapshots \mathbf{s}_j . The dissimilarity between the input data and the representatives has a closed-form expression, but this is not the case with the Grassmann clustering. Indeed, the medoids are reduced bases and the Grassmann distance is only defined between the subspaces they span. As a result of the clustering using the Grassmann distance, each preliminary reduced basis has a label and each preliminary reduced basis is related to a preliminary cluster of data. Hence, we assign to the data in a preliminary cluster the label of its preliminary reduced basis.

4.4.2 Optimal relabelling

Whilst labelling the existing simulation snapshots is possible simply by identifying which snapshots were used for the construction of each reduced basis, the assignment of a cluster to new data is a more difficult task. The previous subsection showed how to label the existing simulation snapshots and introduced a way of assigning a label to new data in the form of partially observed temperature fields. An alternative to these assignment functions is the use of a classification neural network that can link the observed data to a set of labels. The computationally expensive training phase leads to a fast assignment of a label. The universality of this approach relies on optimally labelling the data without the metrics that were used in the clustering offline stage. This can be achieved by computing labels l_j^* taking into account the reconstruction error of the PPCA models Φ^k and the snapshots \mathbf{s}_j :

$$l_j^* = \arg \min_{k \in \llbracket 1, K \rrbracket} \delta_R(\mathbf{s}_j, \Phi^k) \quad (4.21)$$

where δ_R is the reconstruction error with a PPCA model, which is given by:

$$\delta_R(\mathbf{s}_j, \Phi^k) = \|\mathbf{s}_j - (\bar{\mathbf{s}}_j + \Phi^k ((\Phi^k)^T \Phi^k)^{-1} (\Phi^k)^T (\mathbf{s}_j - \bar{\mathbf{s}}_j))\|_2 \quad (4.22)$$

The reconstruction error, similar to the Projection error, measures the error between a snapshot and the orthogonal projection with a PPCA basis, as defined in Chapter 3. This metric is directly connected to the second step of the online stage: the estimation of the 3D temperature and maximum principal stress fields, and

the estimation of the unknown parameters. It offers an optimal assignment of local surrogate models independently of the clustering method used for their construction.

There are significant differences between the labels assigned directly via clustering and the new set of labels defined in Equation (4.21). Table 4.4 sums up how many labels were changed when using this new formulation.

TABLE 4.4: Amount of labels that were changed when comparing the clustering labels and the optimal labels with respect to the PPCA reconstruction error.

	Euclidean	Grassmann	Projection
Total amount	3408	12421	4993
Percentage	10.20 %	37.18 %	14.95 %

These labels are considered optimal in the sense of the reconstruction error, which means that the local reduced model associated to the assigned class is the one that best represents the snapshot. The labels will be used as a way of identifying the "true" class of every snapshot in the training of a classification neural network.

4.4.3 Multi Layer Perceptrons

The selected architecture for the classification neural network is the Multi-Layer Perceptron (MLP) [56]. MLPs are known to be universal approximators [76]. According to Hagan in [65], two hidden layers are sufficient to approximate any function arbitrarily well, according to, provided that enough hidden units are available.

An MLP can be described as the successive application of functional transformations to the input data. Each layer creates new features from a linear combination of the current layer's inputs and applies a non-linear activation function. It is composed of at least an input layer, a hidden layer and an output layer. The number of hidden layers is one of the hyper-parameters of the model. To explain how an MLP works, let us see an example of a single hidden layer model, shown in Figure 4.7 .

Let \mathbf{x} be a p -dimensional input vector and $\mathbf{\Lambda}_k \in \mathbb{R}^K$ be a vector of zeroes except for a 1 at the k^{th} position, which represents its "true label" in a K -class classification problem. The input vector \mathbf{x} is transformed into a vector $\mathbf{y} \in \mathbb{R}^M$ as:

$$\mathbf{y} = f(\boldsymbol{\alpha}_0 + \mathbf{A}\mathbf{x}) \quad (4.23)$$

where $\boldsymbol{\alpha}_0 \in \mathbb{R}^M$ and $\mathbf{A} \in \mathbb{R}^{M \times p}$ are, respectively, a vector and a matrix of coefficients. These coefficients are unknown parameters of the model and are also known as *weights*. Finally, $f : \mathbb{R} \rightarrow \mathbb{R}$ is a derivable non-linear activation function. This function is applied element-wise. Some examples of activation functions are:

- ReLU function: $f(x) = \max(0, x)$
- Sigmoid function: $f(x) = \frac{1}{1+e^{-x}}$
- Hyperbolic tangent: $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

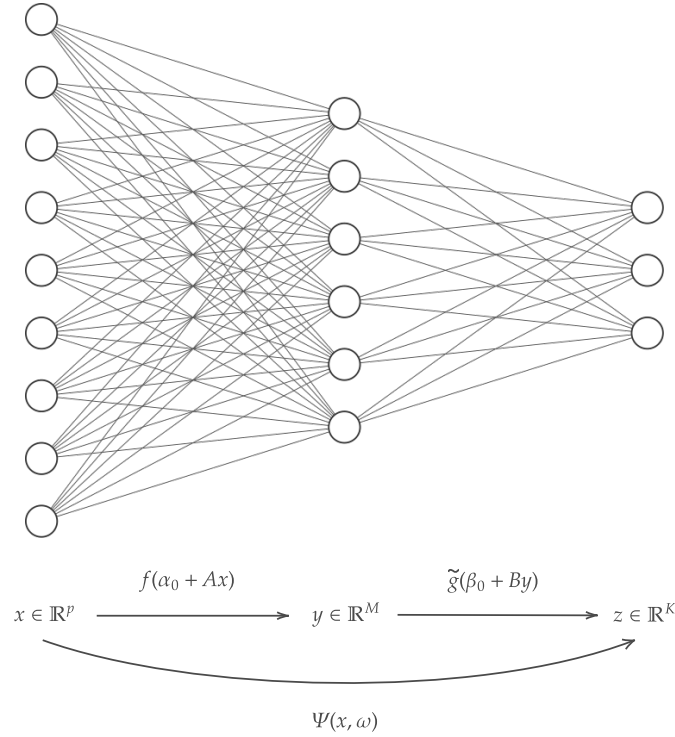


FIGURE 4.7: Scheme of a one hidden layer MLP.

The result of the output layer is a vector $\mathbf{z} \in \mathbb{R}^K$ that is obtained by a linear combination of the elements in \mathbf{y} , and the application of a softmax function g , which is a generalisation of the logistic regression function and outputs positive values that add up to one. Therefore its results can be interpreted as a probability distribution. The softmax function in a K-class classification problem is applied to a K-dimensional vector \mathbf{v} , where the transformation of each of its components is given by:

$$g(v_k) = \frac{e^{v_k}}{\sum_{i=1}^K e^{v_i}} \quad (4.24)$$

The softmax function can be substituted by a log-softmax, which consists in applying the logarithm of the softmax function. The result, in this case, is interpreted as a vector of log probabilities. Using log probabilities offer several advantages in terms of numerical efficiency and stability. The log-softmax function is obtained by applying the natural logarithm to the results of the softmax function:

$$\tilde{g}(v_k) = \log(g(v_k)) = \log \left(\frac{e^{v_k}}{\sum_{i=1}^K e^{v_i}} \right) \quad (4.25)$$

Thus, the vector \mathbf{z} is calculated as:

$$\mathbf{z} = \tilde{g}(\boldsymbol{\beta}_0 + \mathbf{B}\mathbf{y}) \quad (4.26)$$

where $\boldsymbol{\beta}_0 \in \mathbb{R}^K$ and $\mathbf{B} \in \mathbb{R}^{M \times K}$ are another vector and matrix of unknown parameters.

The action of the consecutive layers could be summed up by the application of a non-linear function Ψ that depends on the $M(p+1) + K(M+1)$ weights: $\boldsymbol{\alpha}_0$, \mathbf{A} , $\boldsymbol{\beta}_0$ and \mathbf{B} . In order to simplify the notations, the weights are grouped together in a variable $\boldsymbol{\omega}$. This function Ψ is:

$$\begin{aligned} \Psi : \mathbb{R}^p \times \mathbb{R}^{M(p+1)+K(M+1)} &\longrightarrow \mathbb{R}^K \\ \mathbf{x} &\longrightarrow \Psi(\mathbf{x}, \boldsymbol{\omega}) = \mathbf{z} \end{aligned} \quad (4.27)$$

where \mathbf{z} is calculated as the composition of Equations (4.23) and (4.26). For simplicity's sake, dependency on the weights is dropped from the notation in the following paragraphs.

Finally, the predicted class l for the input vector \mathbf{x} is given by the most probable class:

$$l = \arg \max_{k=1, \dots, K} \Psi(\mathbf{x}) \quad (4.28)$$

The whole procedure can be interpreted as a classifier function \mathcal{C} that assigns a label l to the original input data \mathbf{X} by picking the most probable class, as determined by Ψ :

$$\begin{aligned} \mathcal{C} : \mathbb{R}^p &\longrightarrow \{1, \dots, K\} \\ \mathbf{x} &\longrightarrow \mathcal{C}(\Psi(\mathbf{x})) = l \end{aligned} \quad (4.29)$$

Training the neural network consists of calibrating the weights $\boldsymbol{\omega}$. These parameters are optimised according to an objective function, which in multi-class classification usually is the cross-entropy function. Optimising the cross-entropy loss function is equivalent to maximising the likelihood of the correct label [69, 23]. The cross-entropy depends on the weights and the true label $\boldsymbol{\Lambda}_k$:

$$CE(\boldsymbol{\omega}, \boldsymbol{\Lambda}_k) = - \sum_{k=1}^K \Lambda_k (\Psi(\mathbf{x}))_k \quad (4.30)$$

which is the negative log-probability of the class in $\boldsymbol{\Lambda}_k$.

Training is done in batches of N_{batch} input vectors. The problem should be generalised for a matrix $\mathbf{X} \in \mathbb{R}^{N_{batch} \times p}$ and the true labels are stored in a matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{N_{batch} \times K}$, where each row is a vector such as $\boldsymbol{\Lambda}_k$, where the only non-zero component is a 1 for the true label. When considering a batch, \mathbf{x} should be interpreted as the transpose of a row \mathbf{X}_n , $n = 1, \dots, N_{batch}$ of \mathbf{X} in all previous equations. The cross-entropy is now calculated as the mean cross-entropy of the batch:

$$CE(\boldsymbol{\omega}, \boldsymbol{\Lambda}) = - \frac{1}{N_{batch}} \sum_{n=1}^{N_{batch}} \sum_{k=1}^K \Lambda_{nk} (\Psi(\mathbf{X}_n, \boldsymbol{\omega}))_k \quad (4.31)$$

where $\Psi(\mathbf{X}_n, \boldsymbol{\omega})_k$ denotes the k^{th} component of the neural network's output.

The training process is divided into several steps called *epochs*. In each epoch, all the training set is processed in a two-step process:

- a *forward pass* in which the current weights are used to produce a prediction;
- and a *backward pass* in which the objective function is minimised.

This process is called *back-propagation*. The objective function is minimised with a gradient descent algorithm [116], usually some implementation of stochastic gradient descent.

4.4.4 Training a classification Multi Layer Perceptron

Several architectures of MLPs were tested for assigning a local PPCA model to the vectors of partially observed temperatures and known parameters. The version presented in this document is a 2 hidden layer model with ReLU activation functions (presented in the previous section and in Equation (4.32)). Each layer is followed by a dropout layer to avoid overfitting [155]. The dropout layer sets a probability p of deactivation to each node, which means that the input temporarily set to 0. the dropout layers were set to a probability of $p = \frac{1}{2}$.

$$f(x) = \max(0, x) \quad (4.32)$$

The objective function is the cross-entropy function, minimised with an ADAMW algorithm [84] with warm restarts [97, 96]. It was shown in previous sections that the different clustering metrics produce clusters of varying sizes. The optimal labels introduced in Equation (4.21) also produce an imbalance in the sense of the number of snapshots assigned to each model. Figure 4.8 show the label spreading, i.e. how many snapshots have been assigned for each label. This imbalance in the data distribution is translated by a weight w_k attributed to each cluster:

$$w_k = \frac{\#C_k}{\#C_{k^*}} \quad (4.33)$$

where $\#C_{k^*}$ is the amount of elements in the biggest cluster and serves as a normalisation term. In order to take these weights into account, the formulation of the Cross-entropy objective function presented in Equation (4.34) needs to be changed to:

$$CE(\boldsymbol{\omega}, \boldsymbol{\Lambda}) = -\frac{1}{\sum_{n=1}^N \sum_{k=1}^K w_k \Lambda_{nk}} \sum_{n=1}^N \sum_{k=1}^K w_k \Lambda_{nk} \Psi(X_n, \boldsymbol{\omega})_k \quad (4.34)$$

The warm restarts strategy aims at improving the accuracy of the neural network by setting the learning rate to a new value in each epoch following a cosine annealing schedule and restarting the parameters of the optimiser after a certain number of epochs. It was first proposed by Loshilov [97] for the Stochastic Gradient Descent algorithm. The cosine annealing function is:

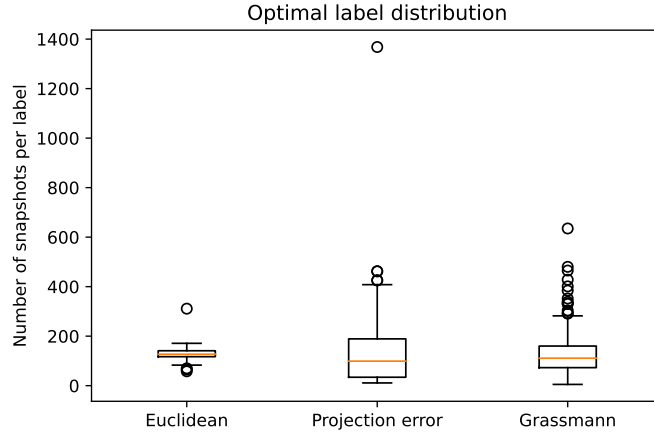


FIGURE 4.8: Optimal label boxplots showing the number of snapshots per label for each clustering metric.

$$\rho_t = \rho_{min} + \frac{1}{2}(\rho_{max} - \rho_{min}) \left(1 + \cos \left(\frac{T_{cur}}{T_i} \pi \right) \right) \quad (4.35)$$

where ρ_t is the learning rate for epoch t , T_{cur} is the number of epochs since the last restart and T_i is the number of epochs between warm restarts. When $T_{cur} = T_i$, the current learning rate ρ_t is set as the minimum learning rate $\rho_t = \rho_{min}$. After a restart, $T_{cur} = 0$ and ρ_t is set to the maximum learning rate $\rho_t = \rho_{max}$. The number of epochs between two restarts can be modified by a multiplier T_{mult} . Thus, the value of T_i is updated after each restart as:

$$T_i \leftarrow T_i \times T_{mult} \quad (4.36)$$

which means that restarts are less frequent over time if $T_{mult} \neq 1$.

One MLP is trained for each clustering method using the PyTorch library [118]. The snapshot database is divided into a training set and a test set with a 90-10 random split. The input data are temperature snapshots \mathbf{T}_j , $j \in \mathcal{J}$, which are extracted from the multiphysics vector \mathbf{s}_j as shown in Equation (4.14). The data had been previously clustered into 260 clusters for the Euclidean and Grassmann metrics and 255 with the Projection Error metric.

4.5 Numerical results

In the following sections, numerical results related to the online phase are displayed. We will focus on two different types of results: the classification problem and the estimation of temperature and stress fields using the local PPCA models created after clustering.

4.5.1 Training of the MLP

The input data is a filtered multiphysics snapshot containing the nodal temperature values on the observable surface of the specimen and the corresponding known parameters μ . The data is randomly divided into a training and test sets with a 90-10 split. Usually, data is pre-processed before training in order to increase performance. This pre-processing generally involves normalisation and scaling of the data. In the present case, the whole dataset does not fit into memory at the same time. As a consequence, the mean and standard deviation of the data have been obtained using Welford's algorithm [170] on consecutive batches. This was tried on all learning strategies presented in the following sections, but the networks didn't improve their accuracy score. As a result, the data was not normalised.

The classifier's output is a numerical label $l \in \llbracket 1, K \rrbracket$ corresponding to the most likely label according to the MLP. In order to quantify the performance of the trained MLPs, the accuracy score is used. The accuracy measures how many labels are correctly assigned with respect to the optimal labels obtained by reconstruction with each cluster's associated basis, assigned as in Equation (4.21). The accuracy score A is computed as:

$$A = \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{\{l_j^* = \mathcal{C}(T^j)\}} \quad (4.37)$$

where N is the number of tested snapshots, l_j^* are the true optimal labels and \mathcal{C} is the classification function. In the following paragraphs and sections, N should be interpreted as the amount of snapshots in the training or test sets when the results refer to either of those sets.

We have tested three different training strategies:

1. Learning rate of 10^{-5} and no warm restarts.
2. Learning rate between $\rho_{max} = 10^{-5}$ and $\rho_{min} = 10^{-8}$, and warm restarts with $T_i = 1$ and a multiplier $T_{mult} = 2$.
3. Learning rate between $\rho_{max} = 10^{-5}$ and $\rho_{min} = 10^{-8}$, and warm restarts with $T_i = 10$ and a multiplier $T_{mult} = 2$.

The learning rate evolution with strategies 2 and 3 are shown in Figures 4.9. The accuracy result of each strategy is shown for each clustering metric in the following subsections. All the networks were trained on a single V100 16Gb Nvidia GPU graphic card on a scientific cluster. The training time for 1000 epochs oscillates between two and three days.

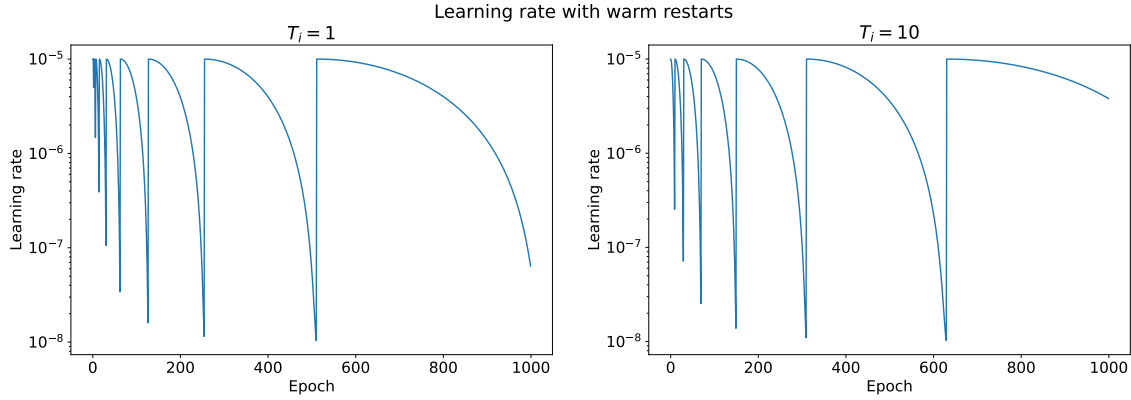


FIGURE 4.9: Evolution of the learning rate following a cosine schedule with $T_{mult} = 2$, and $T_i = 1$ (left) and $T_i = 10$ (right).

Euclidean clustering

The accuracy on the training and test sets for the three training processes are shown in Figures 4.10, 4.11 and 4.12. These figures show that the attempts at improving the accuracy by resetting the learning rates have worked as intended. Whilst the accuracy without Warm Restarts remains below 50% for the test set (see Figure 4.10), the other two strategies go well beyond 80%.

In the cases where the Warm Restarts are used, one can observe a decline in accuracy followed by a steady improvement in the training and test sets for both configurations $T_i = 1$ and $T_i = 10$. Those are the instants where the optimiser parameters and the learning rate are reset. There is no significant difference between both strategies, both of them showing accuracy scores of around 92% for the training set and around 86% for the test set.

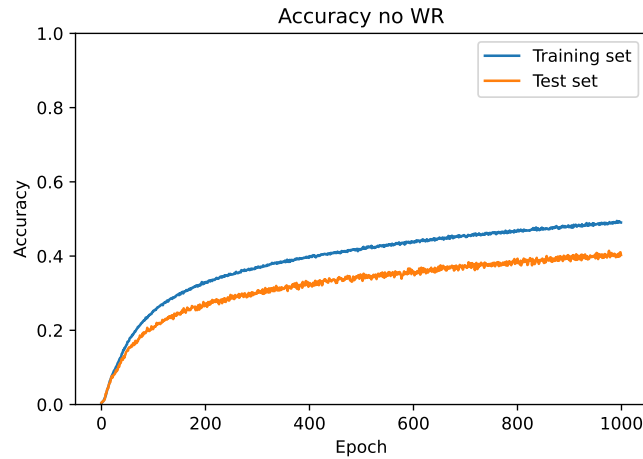


FIGURE 4.10: Evolution of the accuracy over 1000 epochs. No warm restarts.

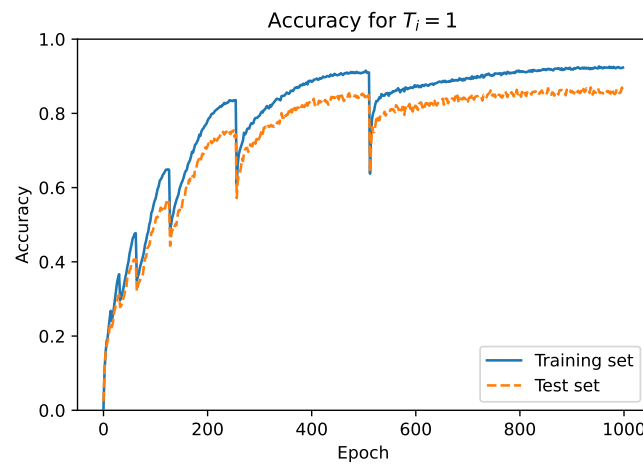


FIGURE 4.11: Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 1$.

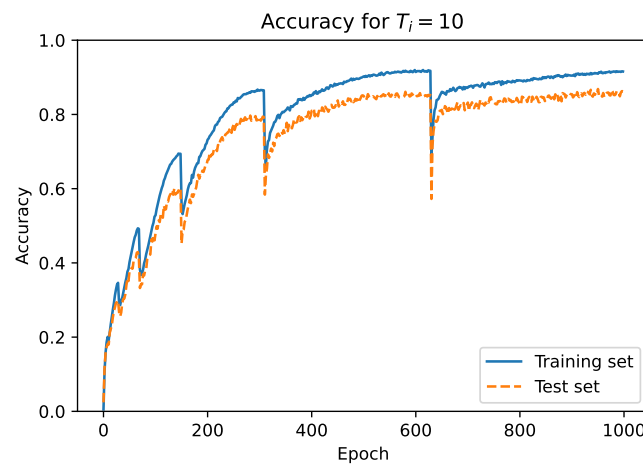


FIGURE 4.12: Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 10$.

Grassmann clustering

The Grassmann clustering proved to be more challenging than the Euclidean clustering. The best results in this case do not go as high as for the Euclidean clustering, with an accuracy between 82 and 85 % with Warm Restarts in the training test and around 70% in the test set (see Figures 4.14 and 4.15). There is a noticeable difference between the training and test sets that was less important for the Euclidean clustering.

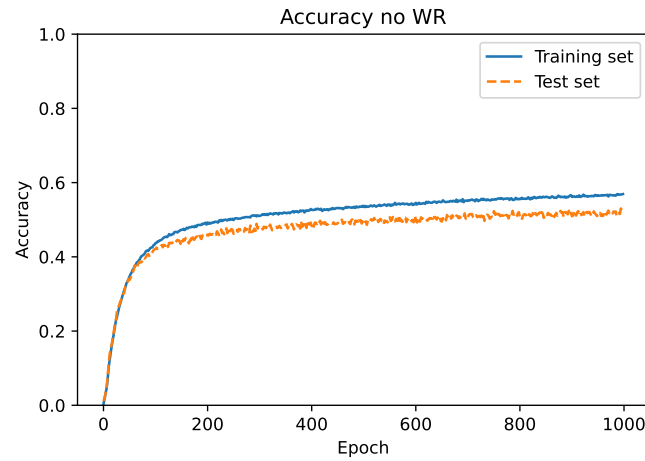


FIGURE 4.13: Evolution of the accuracy over 1000 epochs. No warm restarts.

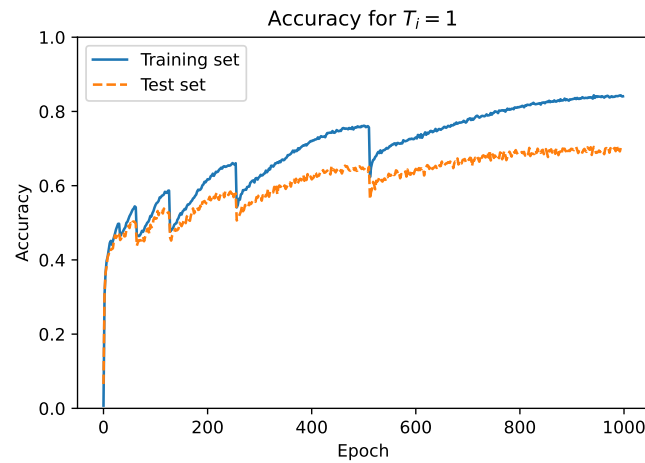


FIGURE 4.14: Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 1$.

When it comes to the training without Warm Restarts, it behaves in the same way as the previous case. The accuracy barely goes over 50%, in particular 56.73% for the training set and 51.51% for the test set, as presented in Figure 4.13. Once again, this proves that the Warm Restarts are helpful with training processes that would get stuck otherwise.

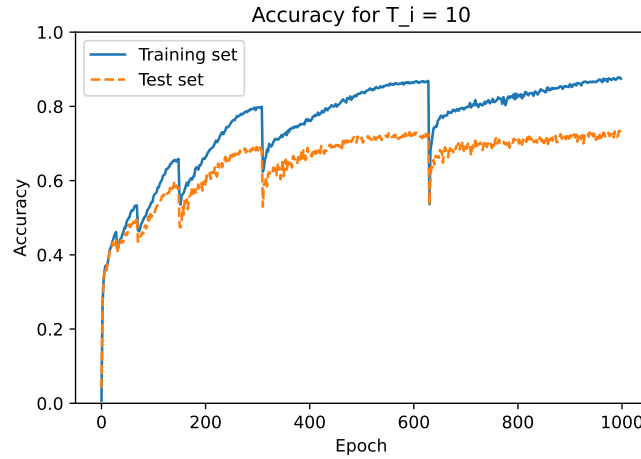


FIGURE 4.15: Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 10$.

Projection error clustering

Finally, the Projection Error clustering has very similar results to the other clustering methods except for a higher accuracy without Warm Restarts. In this case, as it is shown in Figure 4.16, the training and test sets' accuracy reach 70.50% and 68.48% respectively, which is significant when compared to the other two results that were around 50%. We could find no clear explanation to why the Projection error clustering gives a better result than the other two clusterings without Warm Restarts.

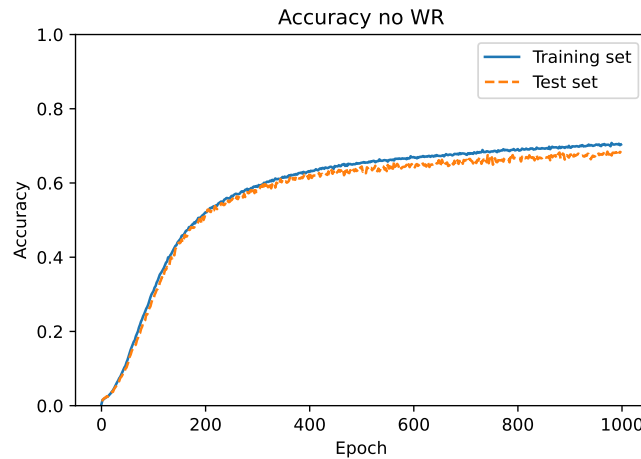


FIGURE 4.16: Evolution of the accuracy over 1000 epochs. No warm restarts.

Both Warm Restart strategies yield very similar results to the ones observed for the Euclidean clustering, but with a different tendency during the training process. Indeed, as it can be observed in Figures 4.17 and 4.18, the accuracy spikes very early

on and then slightly improves over the course of the following epochs. The fact that this fast initial convergence is not continued later does not change the end results, which are again above 80% for the training and test sets in both configurations $T_i = 1$ and $T_i = 10$. Similarly to the Euclidean clustering, the training and test sets' accuracy are very similar, with a training set accuracy close to 88% and a test set accuracy between 82% and 83%.

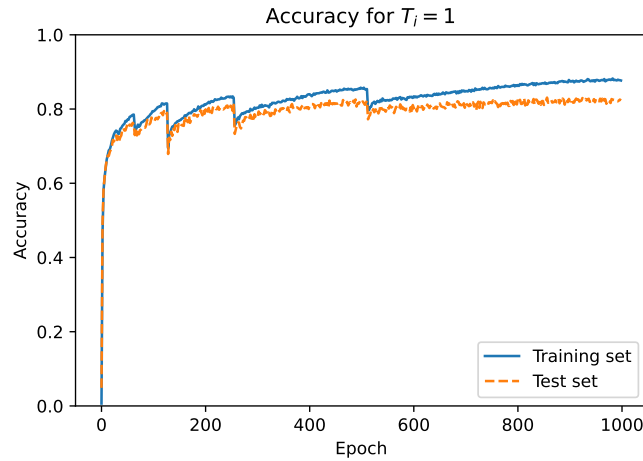


FIGURE 4.17: Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 1$.

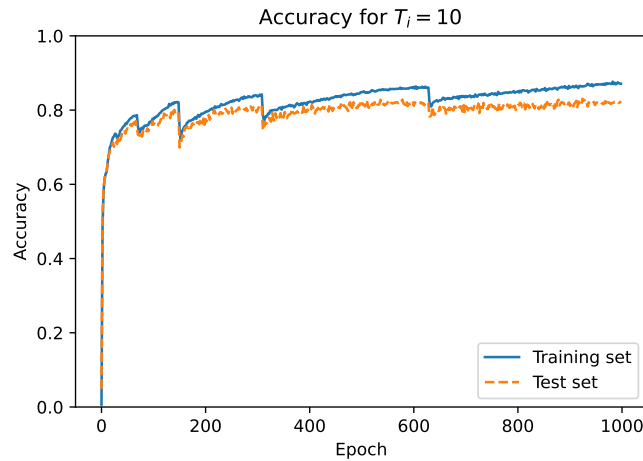


FIGURE 4.18: Evolution of the accuracy over 1000 epochs. Warm restarts with $T_i = 10$.

Table 4.5 sums up the final training accuracy results for the three training strategies. The final accuracy scores shown in the Table confirm what was observed in the previous Figures: there are no major differences between $T_i = 1$ and $T_i = 10$ when using Warm Restarts. The most significant difference appears for the Grassmann clustering, with a difference of around 3 % in favour of $T_i = 10$. In the next

section, only the MLPs trained with Warm Restarts and $T_i = 10$ are used to test the estimation of multiphysics state vectors with the new local bases.

TABLE 4.5: Accuracy results for each MLP and all training strategies.

	Set	Euclidean	Grassmann	Projection
No WR	Training	49.04 %	56.73 %	70.50 %
	Test	40.29 %	51.51 %	68.48 %
$T_i = 1$	Training	92.66 %	84.17 %	88.00 %
	Test	86.35 %	70.73 %	82.85 %
$T_i = 10$	Training	91.83 %	87.27 %	87.63 %
	Test	86.17 %	73.24 %	82.22 %

A high accuracy score is needed to ensure that the input data is correctly assigned the optimal label and, thus, the optimal local PPCA model. Failing at this, the local PPCA model might not be able to estimate the desired physical fields from the observed data.

4.5.2 3D temperature and stress estimation with noisy synthetic data

The numerical tests' objective is to showcase the online phase, starting with the label assignment and then estimating 3D temperature and principal stress fields. White noise is added to each input snapshot with the same variance as the camera error estimated in chapter 2. This error was calculated as the variance between the infrared camera and thermocouple temperature measures. It was estimated to be $\sigma_M^2 = 323.880246$, which corresponds to a standard deviation of $18^\circ C$.

This will produce additional miss-classification by the neural network, as it will be shown. The tests are done on the scientific cluster following these steps:

Numerical tests' steps

1. Select a batch of snapshots.
2. Add white noise to the snapshots.
3. For each clustering method, assign a label using the corresponding trained MLP.
4. Estimate temperature and principal stress fields for each clustering method.
5. Compute relative L^2 error with respect to a reference FE simulation.

The snapshots are selected in batches. Reusing the notation of chapter 3, each vector in the batch is called d_j , $j = 1, \dots, N_S$ and is calculated as:

$$d_j = Hs_j + \epsilon_j \quad (4.38)$$

where $\epsilon_j \sim \mathcal{N}(0, \sigma_M^2)$ is white noise, \mathbf{s}_j is a multiphysics vector and H is a boolean operator that extracts the observed temperature nodes and the known parameters. For these numerical tests, the observed temperature nodes correspond to the entire surface observable surface, the one opposite to the welding arc.

The classifier \mathcal{C} assigns a label $l_j = \mathcal{C}(d_j)$ to the snapshot. This will determine which PPCA model is used in order to obtain the estimations. The estimation is the mean of the posterior distribution $m_{s_l|d_j}^l$. We refer to chapter 3 for all the details on the formulation and the resolution.

The relative L^2 errors were already introduced in chapter 3, in particular in Equations (3.42) and (3.43). In this case, they write:

$$e_T^j = \frac{\|\mathcal{T}(m_{s_l|d_j}^l) - T_{|\mu,\theta}^j\|_2}{\|T_{|\mu,\theta}^j\|_2} \quad (4.39)$$

$$e_{\sigma_I}^j = \frac{\|\sigma_{\mathcal{I}}(m_{s_l|d_j,\mu}^l) - \sigma_{I|\mu,\theta}^j\|_2}{\|\sigma_{I|\mu,\theta}^j\|_2} \quad (4.40)$$

where \mathcal{T} and $\sigma_{\mathcal{I}}$ are operators that extract the entire temperature or principal stress fields respectively.

The temperature and principal stress estimation errors are shown in Tables 4.6 and 4.7 respectively. The tables compare the average estimation error according to different criteria for each clustering metric.

The first two rows show the overall average estimation error for the training and test sets snapshots. This error is averaged over the entire parameter study, independently of whether the classification was successful or not with respect to the true label. The difference between the training and test set's errors are not significant when comparing the same metric for either temperature and principal stress. The difference between metrics is, however, noticeable. The projection error clustering presents the lowest estimation errors in temperature and principal stress, with less than 2 % and 6 %, respectively. On the contrary, the Euclidean clustering obtained the worst results on average, with temperature estimation errors above 7 % and principal stress errors above 22 %.

TABLE 4.6: Average temperature estimation error.

	Set	Euclidean	Grassmann	Projection
Overall	Training	7.4 %	4.60 %	1.80 %
	Test	7.71 %	4.52 %	1.78 %
Correctly assigned	Training	3.78 %	1.60 %	1.05 %
	Test	3.24 %	1.61 %	1.01 %
Incorrectly assigned	Training	35.16 %	21.73 %	6.28 %
	Test	28.45 %	11.43 %	4.94 %

TABLE 4.7: Average principal stress estimation error.

	Set	Euclidean	Grassmann	Projection
Overall	Training	22.34 %	11.68 %	5.96 %
	Test	22.88 %	11.66 %	5.96 %
Correctly assigned	Training	11.90 %	6.78 %	5.23 %
	Test	9.32 %	6.60 %	5.12 %
Incorrectly assigned	Training	102.80 %	40.07 %	10.41 %
	Test	85.92 %	23.79 %	9.45 %

The results are also separated into two groups according to whether the MLP correctly identified the true label or not. The white noise applied to the input data introduces more miss-classifications than the ones obtained during the training process. Indeed, Table 4.8 shows the difference between the accuracy scores in training and with noisy data. The accuracy is slightly lower with noisy data, with decreases between 2 % and 4 %.

TABLE 4.8: Accuracy results for each MLP with noisy input.

	Set	Euclidean	Grassmann	Projection
Accuracy in training	Training	91.83 %	87.27 %	87.63 %
	Test	86.17 %	73.24 %	82.22 %
Accuracy with noisy data	Training	88.47 %	85.07 %	85.79 %
	Test	82.28 %	70.40 %	80.34 %

The third and fourth rows of Tables 4.6 and 4.7 correspond to the snapshots in the training and test sets that the MLP correctly classified. We observe that the estimation error is almost two times smaller than the one observed when mixing all the snapshots, except for the Projection Error clustering which shows a decrease of less than 1 %. Hence, the estimations are closer to the Finite Elements reference solutions for the three metrics and both physical fields, which is the expected behaviour. The temperature errors are remarkably low for the projection error and the Grassmann clustering, which are close to 1 %. The average Euclidean clustering stress' errors when the snapshot is correctly classified are 11.90 % for the training test and 9.32 % for the test set.

Finally, the last two rows of Tables 4.6 and 4.7 present the results when the MLP did not correctly assign the true optimal label. The average errors are increased for all categories, but it is especially significant for the Euclidean clustering. On average, when the snapshot is not correctly classified, the temperature error with the Euclidean clustering are 35.16 % for the training set and 28.45 % for the test set. The order of these errors is high for the temperature field. What is more, the results are even worse for the average principal stress errors, for which relative errors reach 100 %. The estimations are, thus, not close at all to the reference solutions. On the other hand, the average errors with the Projection Error clustering stay very

low even when the snapshot is miss-classified in both temperature (6.28 % for the training set and 4.94 % for the test set) and principal stress (10.41 % and 9.45 %). In particular, it should be noted that the principal stress errors are similar to the Euclidean clustering errors when the snapshots are *correctly* classified. The Grassmann clustering average errors are also greatly increased, but not as much as the Euclidean clustering errors.

Hence, the local PPCA models created from the Projection Error clustering produce better estimation results and, on average, seems to be less influenced by an error in the classification. The following subsections will show examples of the best and worst estimations.

These results were presented on average, but it is well known that extreme values greatly perturb this measure. Table 4.9 presents the percentage of snapshots whose error is below 1, 5, 10 and 20 % in temperature and stress. For temperature, almost 90 % of the snapshot have an estimation relative error under 5 %, reaching 97 % for the projection error clustering. With respect to the maximum principal stress, results are more heterogeneous. More than 85 % of the snapshots have an error below 20 %, including 98.95 % in the projection error clustering. However, the 10 % threshold shows much variance, with 92.93 % in the projection error clustering, 79 % in the Euclidean clustering, but going as low as 52.74 % for in the Grassmann clustering. Overall, the same conclusions can be reached: the projection error clusters tend to produce the better results in term of L^2 norm between the estimations and the reference snapshots.

TABLE 4.9: Percentage of snapshots with an error below a certain threshold.

	Field	Euclidean	Grassmann	Projection
Below 1 %	Temperature	66.51 %	15.08 %	57.13 %
	Stress	0.02 %	0.76 %	0 %
Below 5 %	Temperature	88.90 %	89.95 %	97.09 %
	Stress	50.06 %	13.78 %	48.58 %
Below 10 %	Temperature	91.73 %	95.78 %	98.63 %
	Stress	79.00 %	52.74 %	92.93 %
Below 20 %	Temperature	95.35 %	97.94 %	99.38 %
	Stress	86.12 %	88.84 %	98.95 %

Examples of best and worst temperature estimations

The best temperature estimations for the three clustering metrics are obtained for snapshots that are assigned their optimal reconstruction label and all of them were part of the training set.

The best temperature estimation obtained with the Euclidean clusters is presented in Figure 4.19. It corresponds to a snapshot where the torch has advanced 112.5 mm and its speed and power are $v = 1.12 \text{ mm/s}$ and 652.03 W. The estimation error is 0.11%. Figure 4.20 shows a comparison between the estimated

temperature field's profile along the welding line and the reference snapshot, with the confidence region of the estimation. The selected cluster's medoid is a snapshot with a torch speed of 1.89 mm/s , a power of 657.6 W and located 107.5 mm after the initial position. One can notice that the power and positions are particularly close, which was already observed in Figure 4.2. The assigned cluster contains 109 snapshots with similar positions and powers, with power values between 646.76 W and 665.89 W , and positions between 101.5 mm and 113.5 mm . The speed is the largest source of variance in this cluster, covering almost all the parameter space, going from 1.12 mm/s to 2.78 mm/s . As a consequence, the variance of the estimate is large on Figure 4.20.

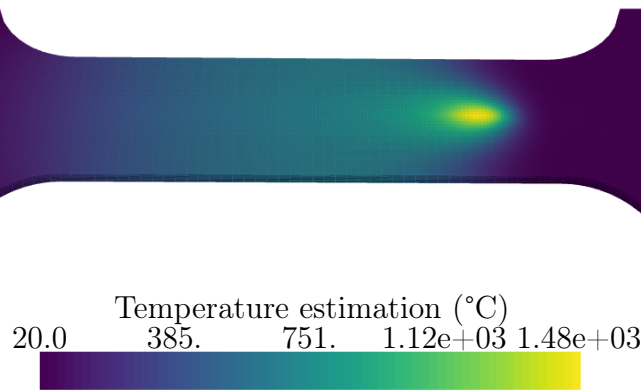


FIGURE 4.19: Best temperature estimation - Euclidean clustering.

In the Grassmann clustering case, the best result (see Figure 4.21) is obtained for a snapshot where the torch is located only 0.5 mm further than its assigned cluster's medoid. The snapshot has the following known parameters $\mu = (2.64 \text{ mm/s}, 843.62 \text{ W})$ and the torch has advanced 82.5 mm , whilst the medoid has the same known parameters μ but the torch has advanced 82 mm . This produces a very small estimation error, at 0.01% , as can be observed in Figure 4.22. The corresponding cluster is cluster 169, which contains only 5 snapshots that are consecutive positions of the same FE simulation, resulting in an estimation with a model with almost no variance.

These snapshots may have been clustered together because of the unknown parameters θ . The unknown parameters vector is $\theta = (11.16, 2.51, 2.36, 0.85, 2.43)$. This vector combines a very large parameter $a_f = 11.16$ and a large coefficient factor $K = 2.43$. These two parameters are responsible for the front and rear lengths of the double ellipsoid in Goldak's equivalent heat source (see Chapter 2 for more details), giving this snapshot a very characteristic look with a very long heat affected zone and low temperature, as shown in Figure 4.21. A similar situation arises in the Projection Error clustering. The best temperature estimation, presented in Figure 4.23, is

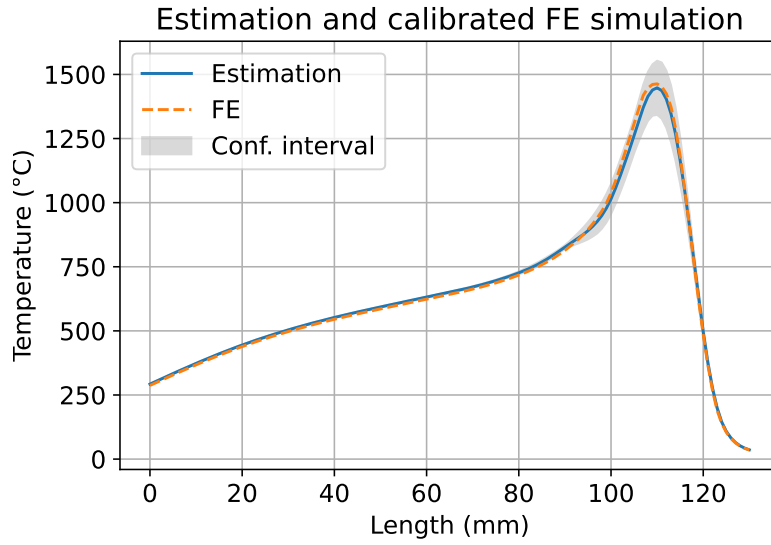


FIGURE 4.20: Best result compared to the reference FE simulation - Euclidean clustering.

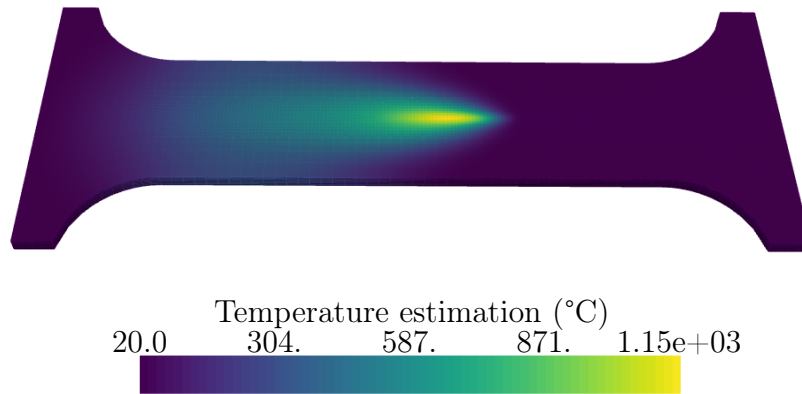


FIGURE 4.21: Best temperature estimation - Grassmann clustering.

obtained for a snapshot with known parameters $\mu = (1.02 \text{ mm/s}, 1071.19 \text{ W})$, which are extreme values of the parameter space. The minimum speed value is 1 mm/s and the maximum power value is 1080 W . The representative of this snapshot's cluster is a snapshot that uses the same known parameters μ . The advancement of the torch is 122 mm for the snapshot and 121 mm for the cluster's representative. The estimation error is, again, very low at 0.14% .

However, in the case of projection error clustering, the assigned cluster is larger

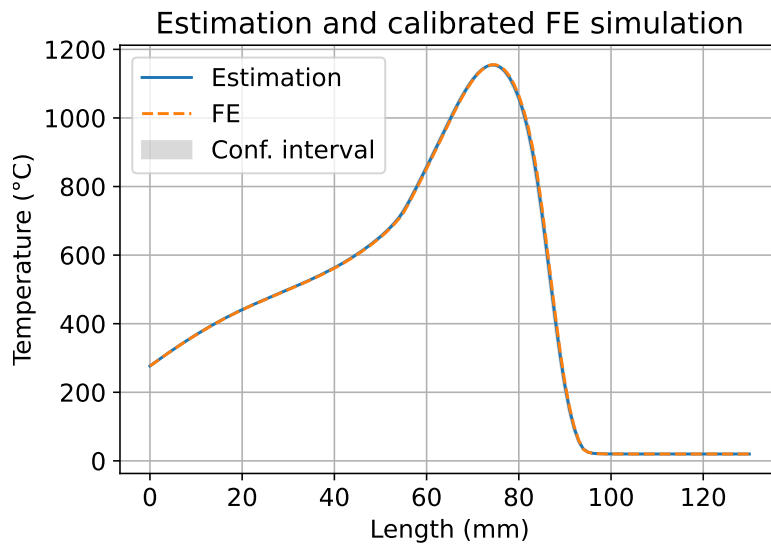


FIGURE 4.22: Best result compared to the reference FE simulation - Grassmann clustering.

than Grassmann's cluster. It contains 91 snapshots with positions between 121 mm and 128.5 mm , with a larger variance in speed and power: speed values between 1.01 mm/s and 1.8 mm/s , and power values between 652.03 W and 1073.83 W . The small confidence region observed in Figure 4.24 indicates, that these large ranges of parameter values result in very similar fields that are accurately captured by the associated PPCA basis.

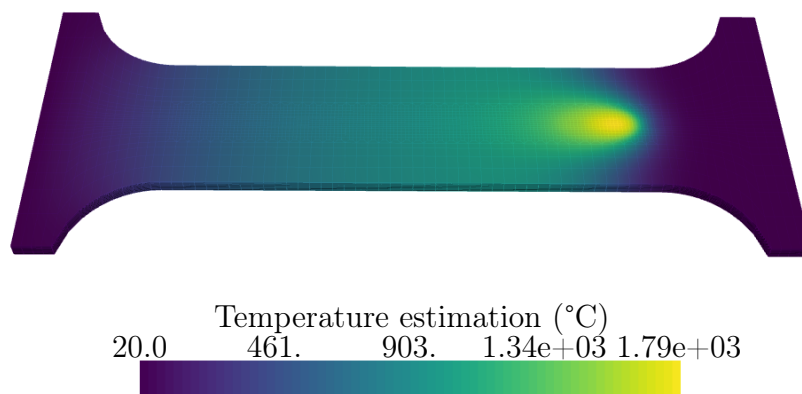


FIGURE 4.23: Best temperature estimation - Projection Error clustering.

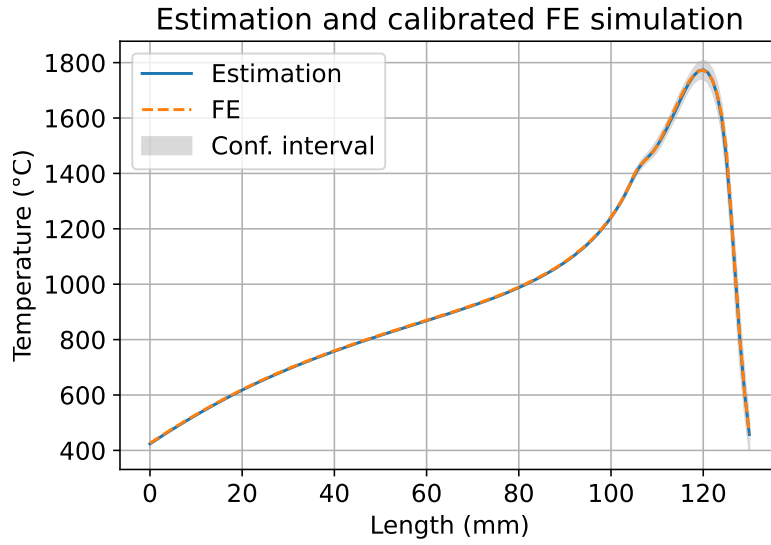


FIGURE 4.24: Best result compared to the reference FE simulation - Projection Error clustering.

The worst results correspond to snapshots that were incorrectly assigned. In some cases, such as the Euclidean cluster case, the snapshot can be assigned to a cluster that cannot properly represent the snapshot. Figure 4.25 shows the temperature estimation of a snapshot with known parameters $\mu = (2.42 \text{ mm/s}, 563.91 \text{ W})$ at position 1.5 mm , which is shown in Figure 4.26. The estimation is not physical, as evidenced by the negative temperature value of 17200°C . This non-physical state estimation is associated to an error of 1458 %.

In the Grassmann clustering case, the classification incorrectly assigned cluster 121 to the snapshot presented in Figure 4.28. This snapshot's speed is 2.44 mm/s , its power is 925.99 W , and its position is 16.5 mm . Cluster 121 is a very small cluster that only contains 6 snapshots, 5 of them being consecutive positions between 16.5 mm and 18.5 mm , but with a known parameters that are quite different from the reference snapshot: $\mu = (1.28 \text{ mm/s}, 637.15 \text{ W})$. This produces a non-physical estimation with an estimation error of 2795 % and with negative values up to -16800°C in some regions, as seen in Figure 4.27.

The Projection Error clustering's worst temperature estimation shows a different situation in which the assigned cluster is concentrated around a very different position with respect to the considered snapshot. Whilst the reference snapshot in Figure 4.30 is located at the end of the weld line, at 113.5 mm , the assigned cluster contains 16 snapshots grouped between positions 3 mm and 5 mm for a large range of speeds (between 1.83 mm/s and 3 mm/s) and powers (between 563.91 W and 1055.56 W). The error between the reference snapshot and the estimation is very large, at 97 %, but much smaller than for the other two clustering methods. However, the estimation, shown in Figure 4.29, is non-physical with negative temperature values, and it does not represent at all the reference snapshot.

These examples have shown some of the advantages and disadvantages of the proposed method. When the snapshots are correctly assigned the optimal label, the associated local models perform very well in the sense of the estimation error.

However, a miss-classified snapshot may try to calculate an estimation with a local basis that is incapable of reproducing the reference snapshot, which might result in a meaningless estimation.

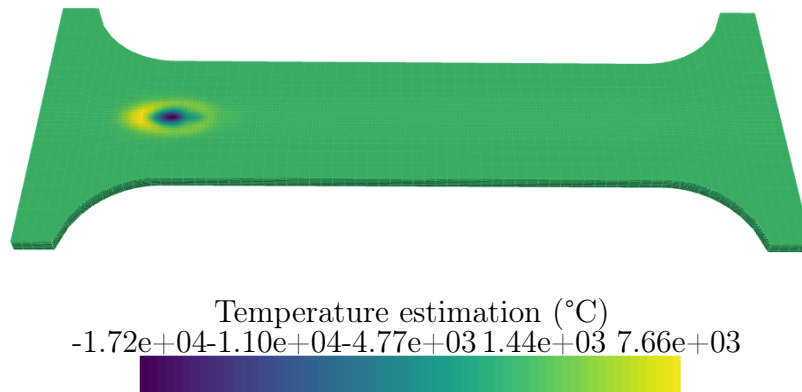


FIGURE 4.25: Worst temperature estimation - Euclidean clustering.

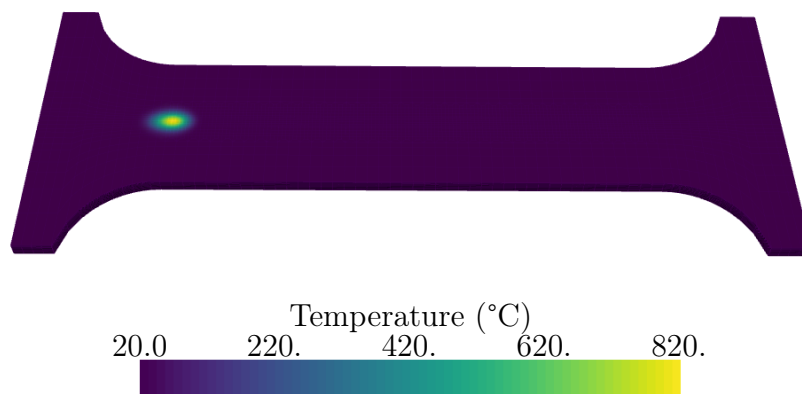


FIGURE 4.26: Reference snapshot - Euclidean clustering.

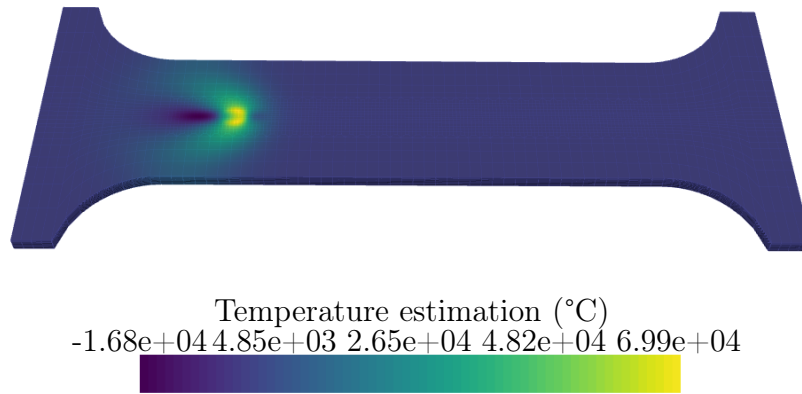


FIGURE 4.27: Worst temperature estimation - Grassmann clustering.

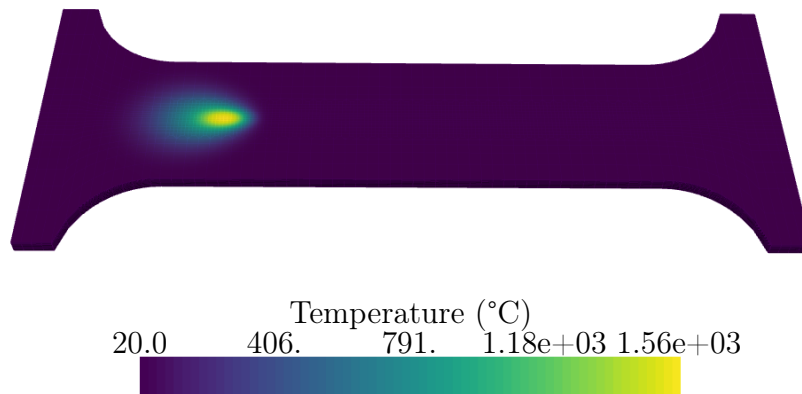


FIGURE 4.28: Reference snapshot - Grassmann clustering.

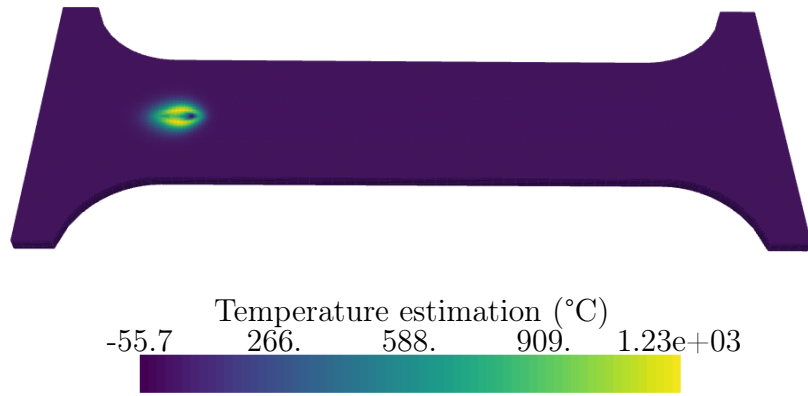


FIGURE 4.29: Worst temperature estimation - Projection Error clustering.

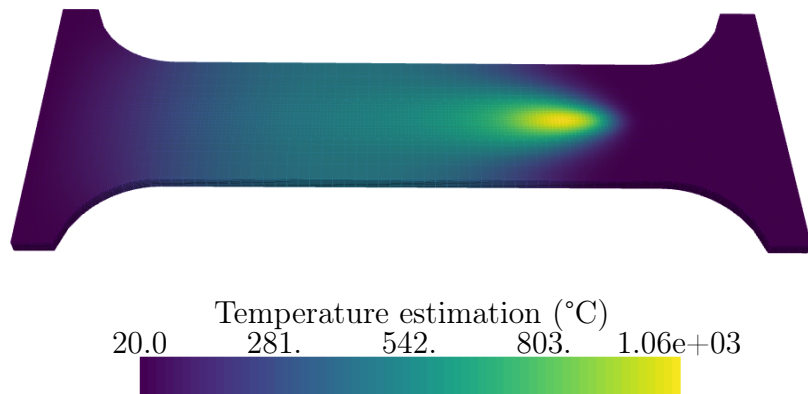


FIGURE 4.30: Reference snapshot - Projection Error clustering.

4.6 Conclusion

This chapter has presented different methods to generate local PPCA models via clustering of a snapshot database. Three metrics that assess dissimilarity between snapshots were proposed. All three of them provide a way of separating the dataset into clusters. Here, the offline stage includes the clustering and the computation of PPCA bases, whilst the online stage refers to the label assignment to a new data point. All of them show advantages and disadvantages during the offline and online stages.

The three proposed metrics are the Euclidean distance, the Grassmann distance and the Projection error between snapshots. However, not all of them treat data in the same way. The main conclusions extracted from the clustering results are :

- The Euclidean distance clustering defined on the space state tends to generate very homogeneous clusters in size and concentrate snapshots around a specific position and torch power values.
- The Grassmann distance clustering defined on a Grassmann manifold, creates heterogeneous clusters in terms of number of snapshots. The clusters contain snapshots grouped mainly through their position on the specimen. The main drawback of this method is that it needs the computation of preliminary local reduced bases.
- The Projection error clustering behaves similarly to the Grassmann clustering in that the clusters are heterogeneous in size and concentrate snapshots according to their position on the specimen. The final number of clusters is not controllable in advance, as the method relies on two alternative parameters: the maximum intracluster error and the minimum size threshold.

All the clustering results share some common properties. The three suggested methods tend to group the data snapshots according to the position of the heat source, which confirms the intuition that led to the partitioning in Chapter 3. Clustering the data snapshots to find an optimal partition is relevant for more complex problems where there is no easily identifiable solution. For problems similar to the PVR experiment, with a linear trajectory of the welding torch, another possibility would be to produce snapshots in the torch reference frame, i.e., in a moving frame along the welding direction.

The first task of the online stage is the assignment of the best possible local model to the input data. Some problems were identified in this context:

- The Grassmann distance is only defined between subspaces spanned by reduced bases, not between snapshots and reduced bases. It cannot be used to determine the closest basis to some new data.
- For large numbers of clusters, the computation of the closest basis might become too expensive. In particular, this could be a problem for the Projection error clustering.

As a universal solution for all three methods, we decided to train a classification model that points towards the best possible model, defined by a new set of optimal labels with respect to the PPCA reconstruction error. The chosen classifier is an MLP neural network. We created one MLP per clustering method and trained them following different training schemes to increase the network's accuracy. In all three cases, the different training schemes resulted in very similar accuracy results for the classifier.

The final results presented in this chapter are the errors between Finite Elements reference solutions and the estimations produced by the PPCA local models. All the snapshots were used as input for the estimation of 3D temperatures and principal stress fields. They were separated into two distinct groups, the training set and the test set (which reference the partition used for the classifier training) and the L^2 relative error was averaged over them. Furthermore, the results were also compared when the classifier correctly assigns the optimal cluster label to the snapshot. This extensive test shows that the PPCA models created from the Projection Error clusters produce the best estimations in all categories, including the cases where the snapshots are miss-classified.

The best and worst estimations presented at the end of the chapter show main advantages and disadvantages of the clustering approach. When a snapshot is correctly assigned, the estimations are very close to the reference snapshot. However, a miss-classification may cause an estimation with a local model that is incapable of correctly reproducing the reference field. This is something that needs to be improved in the future, the objective being that a miss-classification points to a close enough cluster. What is more, even in the case of miss-classification, the estimation of the state should abide by physical laws such as positivity of the temperature fields (measured in K).

Chapter 5

Multipass welding hyper-reduction

Abstract:

Previous chapters have dealt with the PVR experiment, a welding experiment where no filler metal is used. However, most manufacturing and repair operations involve a large number of welding passes. Multipass welding is known to be a challenge for simulations and, in particular, for reduced order models. This chapter presents exploratory work on extensions to the hyper-reduction method in the context of multipass welding. Specifically, two methods are proposed to forecast future states when FE simulations are too long to produce training data for a complete welding operation. Only the first passes are simulated. These results are used to forecast the state of future passes. A hyper-reduced model is built from the forecasted states in order to predict the final state of the weld.

The contents of this chapter have been extracted from an internal report produced at the beginning of the PhD [120] using results from a previous internship.

Résumé :

Les chapitres précédents ont utilisé l'essai de fissuration à chaud PVR comme cas test, un essai de soudage sans métal d'apport. Néanmoins, la plupart des opérations de maintenance et assemblage utilisent une large quantité de cordons de soudure. La simulation d'une opération de soudage multipasses est connue pour sa difficulté, en particulier pour l'application de modèles réduits.

Ce chapitre présente un travail exploratoire sur l'extension de la méthode d'hyper-réduction à un cas de soudage multipasses. En particulier, deux méthodes seront proposées pour prédire l'état thermomécanique futur d'un cordon quand les simulations numériques par Éléments Finis sont trop longues pour produire des données d'entraînement pour l'opération entière. Seulement les premiers cordons seront simulés et ces résultats seront utilisés pour prédire l'état des passes futures. Un modèle hyper-réduit sera construit à partir des prédictions pour réaliser un calcul plus fin de l'état final de la soudure.

Les contenus de ce chapitre ont été adaptés à partir d'un rapport interne produit au début de la thèse [120] en utilisant les résultats d'un stage précédent.

Contents

5.1	Introduction	133
5.2	Simplified multipass welding test case	134
5.3	Physical mechanisms captured by dynamic mode decomposition	134
5.4	Physical fields shift method	137
5.5	Local-global extrapolation method	138
5.6	Inherent strain method	139
5.7	Numerical results	140
5.7.1	Displacement and stress extrapolation	140
5.7.2	Plastic strain extrapolation	142
5.7.3	Hyper-reduced models from extrapolated fields	144
5.7.4	Execution time	147
5.8	Conclusion	148

5.1 Introduction

In welding processes, highly localised transient heat cause non-uniform thermal expansion and contraction and thus result in plastic deformation in the weld and surrounding areas. Then, residual stress, strain and distortion are permanently produced in the welded structures. Stresses are known to promote fracture and fatigue, and no experimental device gives access to the in-situ three-dimensional distribution of stresses. Hence, prediction of residual stresses and distortion for various parameter values of the welding process is essential for optimising a welding process.

Due to the complexity of the physical processes involved in welding, Finite Element Analysis (FEA) play an indispensable role in the numerical simulation of welding. Hibbitt and Marcal [72] presented the first two-dimensional FEA to predict residual stresses in welding. As explained in [177], most FEA simulation efforts during the 70s and 80s were focused on simplified 2D geometries [81], primarily due to computational costs. Since the end of the 80s, 3D welding simulations can be conducted using commercial FEA software, as reported in [156]. In addition, when considering mobile heat sources, numerical simulation has a computational complexity that is nonlinearly growing with respect to the length of the passes involved in the process. On many occasions, the Finite Elements simulations are not feasible in reasonable times.

Recently, projection-based model order reduction has been proposed to speed-up welding simulations in [176, 38] in order to speed-up numerical predictions. In these publications, a reduced basis of empirical modes is extracted from offline simulations before setting the reduced equations for online predictions as explained in [31]. A common way to perform such an extraction of the reduced basis for nonlinear simulations is the POD method proposed in [12]. Although these models have proved to be useful in accelerating non-linear welding simulations, the multipass case is still a challenge. Being an a posteriori method, a reduced model constructed for the n^{th} pass cannot be used to simulate a future bead.

In this chapter, we propose a prediction step to extrapolate finite element solutions from one pass to another in order to obtain a reduced-order model for passes that have not been simulated yet. A complete pipeline of the extrapolation and hyper-reduced computation will be presented.

Two methods to calculate extrapolated simulations are introduced, and then they will be combined to improve their performance. A first extrapolation method uses a dynamic mode decomposition (DMD) of the displacements [146]. Here the DMD describes the underlying physical mechanisms captured in a sequence of displacement fields. We assume that these underlying physical mechanisms can be extended from a set of first passes to the remainder of passes. The second extrapolation method is the Physical Fields Shift Method [121]. In this approach, physical transformations in the neighbourhood of the heat source are assumed to be almost steady-state transformations in a moving frame. A final method combines both previous ones in order to improve the results. The physical domain is divided into two distinct zones with a clustering algorithm and one method is applied on each zone. Results using all the presented approaches and the performance of the extrapolated reduced models are shown in a final section.

5.2 Simplified multipass welding test case

The extrapolation methods will be applied to a simplified case of multipass welding. The test case consists of a 316L stainless steel plate with $N = 20$ deposited beads on top of it (see Figure 5.1). The metal base dimensions are $100 \times 200 \times 20 \text{ mm}$ and each bead's dimensions are $52 \times 4 \times 2 \text{ mm}$. The mesh is a regular grid containing $\mathcal{N} = 57768$ nodes. For details on the numerical model and temperature-dependent material parameters, see Chapter 2. The operational parameters for this simplified case are displayed in Table 5.1.

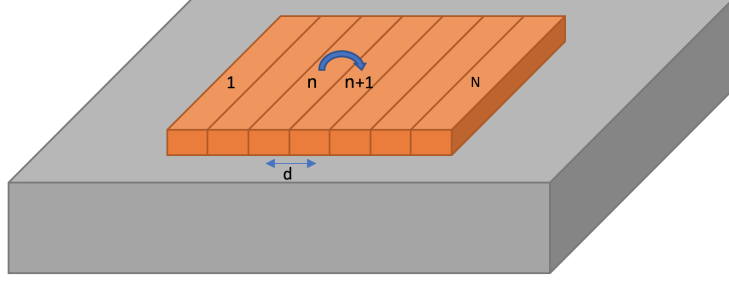


FIGURE 5.1: Simplified scheme of a N passes multipass welding.

TABLE 5.1: Simplified test case operational parameters.

Current	200 A
Voltage	12 V
Travel speed	1.00 mm/s
Linear energy	2400 J/mm

The results of the simulation after each pass n are stored in a matrix $\mathbf{Q}^n \in \mathbb{R}^{\mathcal{N} \times \tau}$, $n = 1, \dots, N$ such that q_{ij} is the nodal value of a quantity of interest at node i and time step t_j , $j = 0, \dots, \tau$. These quantities of interest are defined over a domain Ω that contains the metal base and all the beads. Physical properties are activated as the heat source progresses. Before the deposition of the weld joint, points in the path of the heat source have the properties of the void.

The complete simulation of the multipass welding case has, thus, $\mathcal{T} = N \times \tau$ time steps. The quantities of interest are mechanical fields such as displacement, stress or plastic strain.

5.3 Physical mechanisms captured by dynamic mode decomposition

Dynamic Mode Decomposition (DMD) [146, 24] has been introduced in fluid dynamics to extract dynamic information from flow fields that are numerically simulated or measured in an experiment. The extracted dynamic modes have been interpreted

as a generalisation of global stability modes in view of the analysis of localised instability phenomena. In this document, DMD defines a recurrence relation between displacement fields of consecutive welding passes. In other words, it will represent the evolution of the displacement between a certain pass $n - 1$ and the next pass n , with $n \leq N$, the total number of passes. We will consider that simulation data is restricted solely to the weld's first $n - 1$ first passes. Let us assume, without loss of generality, that the objective quantity of interest is displacement.

After the simulation of $n - 1$ passes, simulation data related to displacement are arranged in a matrix $\mathbf{Q} \in \mathbb{R}^{\mathcal{N} \times (n-1)\tau}$, such that each element q_{ij} of the matrix denotes the nodal value of displacement at the mesh node i , $i = 1, \dots, \mathcal{N}$ and at time step j , $j = 1, \dots, \tau$. The number of columns in \mathbf{Q} increases on the fly after each time step during the FE simulation of the welding process. Here we consider sets of data related to the same pass of welding. We denote $\mathbf{v}_k \in \mathbb{R}^{(\mathcal{N}\tau)}$ the vector of data related to the k^{th} pass:

$$v_k(p) = Q(i, j), \quad p = (j - \tau(k - 1) - 1)\mathcal{N} + i, \quad (5.1)$$

with: $i \in \{1, \dots, \mathcal{N}\}$, $j \in \{\tau(k - 1) + 1, \dots, \tau k\}$.

The set of data related to a group of $n - 1$ passes between pass number one and pass number $n - 1$ has the following matrix form:

$$\mathbf{V}_1^{n-1} = [\mathbf{v}_1, \dots, \mathbf{v}_{n-1}] \in \mathbb{R}^{(\mathcal{N}\tau) \times (n-1)} \quad (5.2)$$

The dynamic mode decomposition aims to identify an empirical linear model to predict $\mathbf{V}_2^n \in \mathbb{R}^{(\mathcal{N}\tau) \times (n-1)}$ knowing \mathbf{V}_1^{n-1} , with:

$$\mathbf{V}_2^n = [\mathbf{v}_2, \dots, \mathbf{v}_n] \in \mathbb{R}^{(\mathcal{N}\tau) \times (n-1)} \quad (5.3)$$

Following Schmid [146], the last vector \mathbf{v}_n can be expressed as a linear combination of the previous ones. In the multipass welding case, this relation assumes that the displacement obtained during a welding pass can be expressed as a linear combination of the displacements related to previous passes.:

$$\mathbf{v}_n = a_1 \mathbf{v}_1 + \dots + a_{n-1} \mathbf{v}_{n-1} + \mathbf{r} \quad (5.4)$$

or in matrix form:

$$\mathbf{v}_n = \mathbf{V}_1^{n-1} \mathbf{a} + \mathbf{r} \quad (5.5)$$

with $\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \in \mathbb{R}^{n-1}$ and $\mathbf{r} \in \mathbb{R}^{\mathcal{N}\tau}$ a residual vector.

Citing Ruhe [139], Schmid then introduces the following matrix expression:

$$\mathbf{A} \mathbf{V}_1^{n-1} = \mathbf{V}_2^n = \mathbf{V}_1^{n-1} \mathbf{C} + \mathbf{r} \mathbf{e}_{n-1}^T \quad (5.6)$$

where $\mathbf{e}_{n-1}^T \in \mathbb{R}^{n-1}$ is the $(n - 1)^{th}$ unit vector. The matrix \mathbf{C} has the following form:

$$\mathbf{C} = \begin{pmatrix} 0 & & & a_1 \\ 1 & 0 & & a_2 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 & a_{n-2} \\ & & & 1 & a_{n-1} \end{pmatrix} \quad (5.7)$$

A robust approach to compute \mathbf{C} involves the SVD of $\mathbf{V}_1^{n-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$. Substituting \mathbf{V}_1^{n-1} in Equation (5.6), we obtain:

$$\mathbf{U}^T \mathbf{A} \mathbf{U} = \mathbf{U}^T \mathbf{V}_2^n \mathbf{W} \mathbf{\Sigma}^{-1} \equiv \tilde{\mathbf{C}} \quad (5.8)$$

The matrix \mathbf{C} contains empirical dynamics information related to the n first passes. In the construction of the DMD, we identify $\tilde{\mathbf{C}}$ as a matrix that allows to compute an approximation of \mathbf{V}_3^{n+1} via the following extrapolation:

$$\mathbf{V}_3^{n+1} \approx \tilde{\mathbf{V}}_3^{n+1} = \mathbf{V}_2^n \tilde{\mathbf{C}} \quad (5.9)$$

An approximation of the displacement of the $(n+1)^{th}$ pass, v_{n+1} is included in the last column of \mathbf{V}_3^{n+1} :

$$\tilde{\mathbf{V}}_3^{n+1} = [\mathbf{v}_3, \dots, \tilde{\mathbf{v}}_{n+1}] \quad (5.10)$$

The extrapolation of matrix \mathbf{Q} is denoted by $\hat{\mathbf{Q}}$ such that:

$$\hat{\mathbf{Q}} = [\mathbf{Q}, \tilde{\mathbf{Q}}], \quad \tilde{\mathbf{Q}} \in \mathbb{R}^{\mathcal{N} \times \tau} \quad (5.11)$$

with the following prediction of the displacement for pass number $n+1$:

$$\tilde{Q}(i, j) = \tilde{v}_{n+1}(p), \quad p = (j-1)\mathcal{N} + i, \quad (5.12)$$

with: $i \in \{1, \dots, \mathcal{N}\}$, $j \in \{1, \dots, \tau\}$

Hence for pass number $n+1$ the following extrapolation holds, for $\theta_j \in \{\tau n + 1, \dots, \tau(n+1)\}$:

$$\tilde{\mathbf{u}}(\mathbf{x}, \theta_j) = \mathbf{u}_e(\mathbf{x}, \theta_j) + \sum_{i=1}^{\mathcal{N}} \varphi_i(\mathbf{x}) \hat{\mathbf{Q}}(i, j) \quad \forall \mathbf{x} \in \Omega \quad (5.13)$$

where the elastic prediction $\mathbf{u}_e(\mathbf{x}, \theta_j)$ takes into account the moving heat source by using a fast simulation.

Schmid shows in [146] that DMD captures the evolution of a dynamical system efficiently. Here, applying a multipass welding case consisting of 10 welding beads deposited on a metallic base gives an accurate prediction over a significant part of the studied domain. The results on the new bead are, however, not satisfactory, as we can see in Figure 5.2. The displacement field for the 10th pass is defined on a new region of the domain after deposition of the bead. We decided to combine this method with the Physical Fields Shift method, presented in Section 5.4, which will be able to predict the behaviour of the new bead.

This happens because the domain changes after the deposition of a new bead in multipass welding.

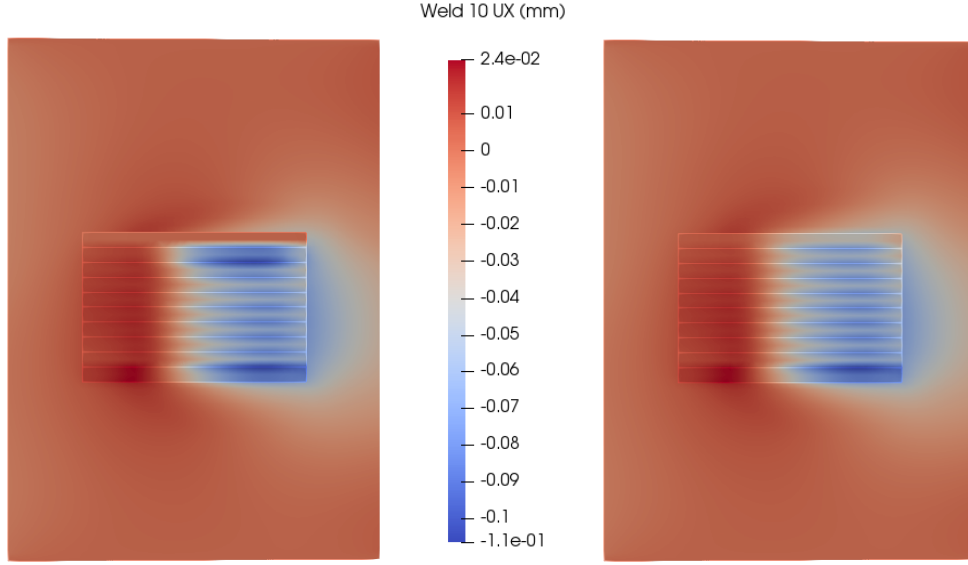


FIGURE 5.2: Example of the extrapolation of the first component of the displacement by using the DMD with $n = 9$, on the left the extrapolated displacement at the end of pass #10, on the right the FE prediction

5.4 Physical fields shift method

The Physical Fields Shift method[121] is a simplified method used in multipass welding simulation to predict new passes from the data of previously computed ones. This method is an early attempt at using extrapolation for the construction of reduced models. It is used in cases where the welding beads have the same geometry, and the deposition of the new beads is done in the same direction. It is based on the similarities observed between two consecutive passes. After a certain number of simulated passes, we observe a propagation of the solution in the direction of the deposition of welding beads. In fact, after each pass, the main difference between two consecutive passes is a spatial translation. Indeed, almost the same fields reappear on the new bead, and on the previous beads, the stress and displacement accumulate. This evolution will be capitalised to obtain an approximation of the subsequent passes.

Let us assume that the first $n < N$ passes of a multipass welding simulation have been computed using the Finite Elements method. We would like to approximate the final state of displacement and stress \mathbf{s}_{n+1} after the $n+1$ pass from the solutions of the n and $n-1$ passes.

An increment field $\Delta \mathbf{s}_n$ is computed to capture the evolution between passes $n-1$ and n :

$$\Delta \mathbf{s}_n = \mathbf{s}_n - \mathbf{s}_{n-1} \quad (5.14)$$

This evolution is assumed to be similar to the one that will be produced between passes n and $n+1$. When the increment field is calculated, we need to shift it of a distance d in the direction of the deposition of welding beads. The shifted

increment field $\Delta \hat{\mathbf{s}}_n$ approximates the evolution that will happen between the m and $m+1$ passes.

The approximation $\tilde{\mathbf{s}}_{n+1}$ of \mathbf{s}_{n+1} is computed as the last simulated pass (\mathbf{s}_n) plus the predicted evolution between the n and $n+1$ passes:

$$\tilde{\mathbf{s}}_{n+1} = \mathbf{s}_n + \Delta \hat{\mathbf{s}}_n \quad (5.15)$$

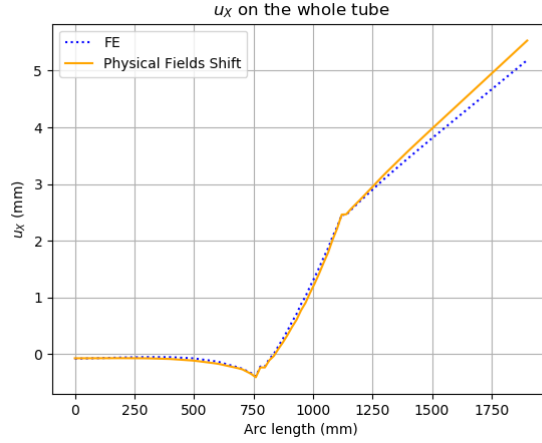


FIGURE 5.3: Approximation of the u_X component of displacement field in an industrial Overlay case.

The prediction is very accurate in the welding area, but far from this zone, this method might produce errors after applying it several times. Figure 5.3 shows an example of the use of the method on an industrial Overlay repair. The value of the displacement component u_X is plotted over a line which goes from one end of the tube to the other. One hundred beads are deposited over the tube in an area between 750 and 1150 mm. PFS was applied 80 times to determine the final distortion after the 100 passes.

5.5 Local-global extrapolation method

In the previous sections, two extrapolation methods were presented. Figure 5.2 illustrated that the DMD method does not predict correctly the final passes. Previous work on the Physical Fields Shift showed that the prediction on the first beads suffer from boundary effects. A better prediction might be generated by combining both methods by applying them where they work best. We propose to separate a region where the moving heat source has a strong effect such that the Physical Fields Shift method is accurate from a region where DMD should be accurate. This is done by using clustering the simulation data.

Let us assume that the first n passes are simulated. For each pass i , the final state (at time $t = \tau^i$, $i = 1, \dots, n$), the nodal values of the objective quantity of interest (displacement or stress, for example), are denoted q^i , $i = 1, \dots, n$. The objective is to compute an estimate \tilde{q}^{n+1} of the future state q^{n+1} at the end of the pass $n+1$. In order to evaluate where each method will perform best, we will

compute an estimation \tilde{q}^n of q^n using the Physical Fields Shift method and compare it to the FE solution. For each node of the mesh a relative error is calculated:

$$\eta_i = \frac{|q_i^n - \tilde{q}_i^n|}{\|q^n\|_F}, \quad \forall i \in \mathcal{L} \quad (5.16)$$

where \mathcal{L} is the list of nodes, q_i^n is the value of the field in the node i , $i \in \mathcal{L}$, and \tilde{q}_i^n is the value of the prediction in the node i .

To separate the regions, three methods are proposed:

- Selecting a criterium ε and then creating two sets: $\mathcal{I}_{PFS} = \{i \in \mathcal{L} : \eta_i \leq \varepsilon\}$ and $\mathcal{I}_{DMD} = \{i \in \mathcal{L} : \eta_i > \varepsilon\}$.
- Applying the Kmeans method [94] to create two clusters \mathcal{I}_{PFS} and \mathcal{I}_{DMD} .
- Applying Hierarchical Clustering (Ward's method) [55] to create two clusters \mathcal{I}_{PFS} and \mathcal{I}_{DMD} .

This relative error indicates in what regions the PFS produces an accurate estimation of the last simulated pass. In that area, the PFS will be applied when computing the prediction of the pass $n + 1$.

5.6 Inherent strain method

In order to compute a mechanical hyper-reduced order model, displacement and stress fields are needed to obtain the primal and dual bases, respectively [45]. The methods presented in previous sections can estimate those results, but there is no guarantee that the forecasted fields are mechanically admissible. In order to ensure that, the forecasting will be performed on plastic strain fields and, then, the displacement and stress fields will be calculated in an additional step. This step is based on the inherent strain method, which is a hybrid experimental/analytical method used to determine residual stresses from sectioning and strain measures [163, 162]. Residual stresses in a body appear as a consequence of an incompatible strain field, which is measured experimentally. Then, the Finite Elements method is used to compute the relation between the stress and the strains. Assuming an elastic behaviour, this relation reads:

$$\underline{\underline{\sigma}} = \underline{\underline{C}}(\underline{\underline{\varepsilon}} - \underline{\underline{\varepsilon}}^*) \quad (5.17)$$

with $\underline{\underline{\sigma}}$ the stress, $\underline{\underline{\varepsilon}}$ the total strain, $\underline{\underline{\varepsilon}}^*$ the inherent strain and $\underline{\underline{C}}$ the elastic constitutive tensor.

The inherent strain method has been applied to many different problems; for instance, Hill and Nelson have used it to determine the residual stress in a long welded joint in [75]. Instead of a measured field of inherent strains, an extrapolation of a plastic strain is used. The relation between stress and strain now reads:

$$\underline{\underline{\sigma}} = \underline{\underline{C}}(\underline{\underline{\varepsilon}} - \underline{\underline{\varepsilon}}^p) \quad (5.18)$$

with $\underline{\underline{\varepsilon}}^p$ the plastic strain.

A linear elastic thermo-mechanical FEA is then computed, imposing the predicted strain on each mesh node. The result is an approximation of the displacement and stress fields that will be used to build a hyper-reduced order model.

5.7 Numerical results

In this section, several tests are carried out. First of all, the DMD and PFS methods are used to compute forecasts of the 11th pass of the simplified multipass welding test case. Then, the same test will be applied to the plastic strain field. The final tests will analyse the results of the hyper-reduced model constructed from the forecasted fields. In the following sections, the relative errors are computed by:

$$e_n = \frac{\|q_n - \tilde{q}_n\|_2}{\|q_n\|_2} \quad (5.19)$$

where $q_n \in \mathbb{R}^{N \times \tau}$ is the reference FE solution for the objective quantity of interest and $\tilde{q}_n \in \mathbb{R}^{N \times \tau}$ is the forecast.

5.7.1 Displacement and stress extrapolation

The first tests concern the extrapolation of nodal displacement and stress fields. The test is done as follows:

Displacement and stress extrapolation

1. Compute the first 11 passes of the simplified multipass test case.
2. Use the first 10 results to forecast the state of the 11th pass using the presented methods:
 - (a) DMD
 - (b) PFS
 - (c) Combination ($\varepsilon = 10^{-4}$)
 - (d) Combination ($\varepsilon = 5 \cdot 10^{-4}$)
 - (e) Combination (Kmeans)
 - (f) Combination (Ward)
3. Compare the results to the FE simulation of the 11th pass.

Figures 5.4 and 5.5 illustrate the final snapshot at time step 11τ obtained with two of the tested methods and the FE simulation. From left to right, one can observe the estimation with the combination of DMD and PFS using a criterium $\varepsilon = 5 \cdot 10^{-4}$, the estimation with the combination of DMD and PFS using Ward's Hierarchical Algorithm and the FE solution.

A full comparison of the methods is presented in Table 5.2. The relative error of the displacements field is very low for all methods, being below 2% in all cases. However, the relative error of the nodal stress field is very high for the DMD and PFS methods, at 29% and 13%, respectively. The combinations prove to produce better estimations of the stress field, with Ward's method producing an estimation with an error lower than 2%.

TABLE 5.2: Comparing different methods to predict the 11th pass.

Method	Error	
	Displacements	Stress
DMD	2%	29%
Physical Fields Shift	0.8%	13%
Combination ($\varepsilon = 10^{-4}$)	0.3%	6.1%
Combination ($\varepsilon = 5 \cdot 10^{-4}$)	0.3%	2.9%
Combination (Kmeans)	1.2%	4.4%
Combination (Ward)	1.97%	1.95%

In Figure 5.4, in particular in the Ward prediction, there is a discontinuity in the forecasted displacement field. This represents a non physical behaviour that needs to be avoided. The apparition of such a discontinuity indicates that the predicted field might not be mechanically admissible and which motivates the use of the inherent strains method on the plastic strain in order to produce the displacement and stress.

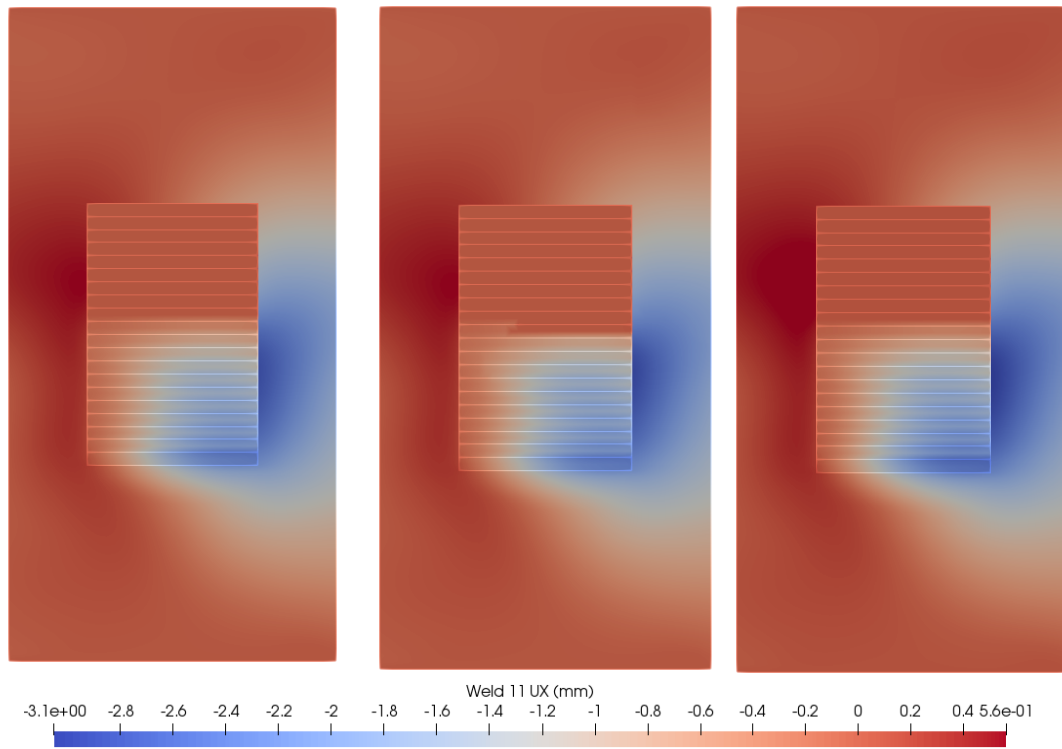


FIGURE 5.4: Comparing the final state of displacements (u_X component) after the 11th pass. (From left to right: $\varepsilon = 5 \cdot 10^{-4}$, Ward, FE simulation).

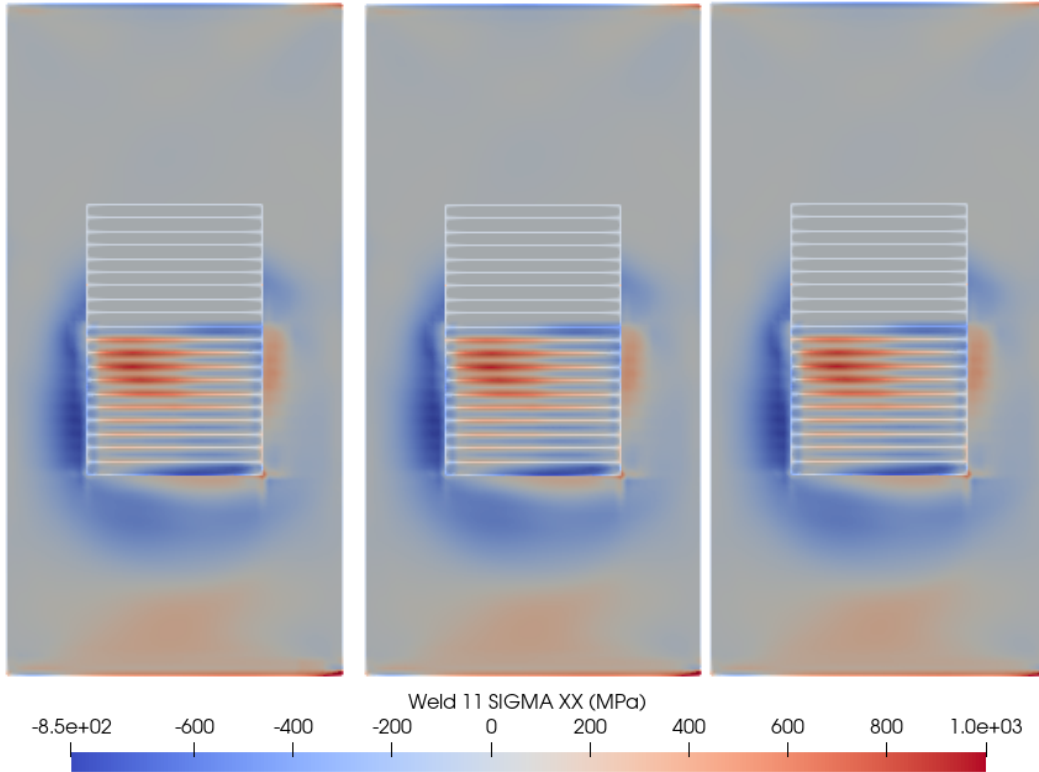


FIGURE 5.5: Comparing the final state of stress (σ_{XX} component) after the 11th pass. (From left to right: $\varepsilon = 5 \cdot 10^{-4}$, Ward, FE simulation).

5.7.2 Plastic strain extrapolation

In the previous section, it was shown that the forecasted states might not be mechanically admissible. In order to avoid that, a new approach based on the prediction of plastic strain and the use of the inherent strain method is proposed. The main idea is to predict the plastic strain suffered by the structure after the $n + 1$ pass and then use the inherent strain method to obtain displacement and stress fields that respect the equilibrium equations.

TABLE 5.3: Comparing different methods to predict the 11th pass' plastic strain.

Method	Error		
	Plastic strain	Displacements	Stress
DMD	27.02%	2.16%	53.6%
Physical Fields Shift	9.38%	0.18%	11.50%
Combination (Kmeans)	23.36%	0.30%	34.12%
Combination (Ward)	23.07%	0.23%	31.16%

The tests are done similarly to the previous ones:

Plastic strain extrapolation

1. Compute the simulation of the first 11 passes of the simplified multipass test case.
2. Use the first 10 results to forecast the state of the plastic strain for the 11th pass using the following methods:
 - (a) DMD
 - (b) PFS
 - (c) Combination (Kmeans)
 - (d) Combination (Ward)
3. Compute the displacement and stress fields using the inherent strains method.
4. Compare the predicted plastic strain, displacement and stress to the FE simulation of the 11th pass.

Table 5.3 reports a summary of the tests. It presents 5 columns. The first one, as in table 5.2, shows the method used to compute the prediction. The second one shows the relative error of the prediction of plastic strain compared to the FE solution. The third and fourth columns represent the relative error of the inherent strain method compared to the FE simulation. The predictions of plastic strain are not as accurate as the displacement and stress ones shown in Table 5.2. In this case, the errors are higher than 20% for all the methods except the PFS. However, the inherent strain method gives a close approximation of the displacement fields despite the high plastic strain errors. The relative errors are remarkably low for all extrapolation methods except for the DMD.

The inherent strains method uses an elastic constitutive law in order to accelerate the computation time of the simulation. This simplification affects the stress fields predictions more than the displacement ones. As a result, the stress extrapolations are not very accurate except for the PFS, with a relative error of 11.5%. The relative error obtained by combining DMD and PFS are above 30%, whilst for the DMD it is higher than 50%.

An additional test is proposed in order to verify if the plastic strain of consecutive passes can be predicted. This would be particularly useful in cases where the number of passes is high and computing the entire simulation is very costly. In this test, the first 17 passes are simulated using the FE method. The goal is to predict the 20th pass. The test is performed as follows:

Plastic strain extrapolation of consecutive passes

1. Compute the simulation of entire simplified multipass test case (20 passes).
2. Use the first 17 results to forecast the state of the plastic strain for the 18th, 19th and 20th passes using the following methods:
 - (a) Combination (Kmeans)
 - (b) Combination (Ward)
3. Compute the displacement and stress fields of the 20th pass using the inherent strains method.
4. Compare the predicted plastic strain, displacement and stress to the FE simulation of the 20th pass.

The results of this test are shown in Table 5.4. The relative error of the plastic strain predictions are higher than the ones in Table 5.3. However, the displacement and stress show similar levels of error. This results are very satisfactory because the hyper-reduced models are built from the displacement and stress fields, and not from the plastic strain. The final tests will deal with the construction of hyper-reduced models.

TABLE 5.4: Comparing different methods to predict the 20th pass' plastic strain.

Method	Error		
	Plastic strain	Displacements	Stress
Combination (Kmeans)	39.82%	0.40%	32.78%
Combination (Ward)	39.77%	0.3%	29.95%

5.7.3 Hyper-reduced models from extrapolated fields

The final tests presented in this chapter consist of creating a hyper-reduced model (HROM) from the forecasted displacement and stress fields. The HROM is computed from a POD basis that covers the entire temporal domain of a pass. The objective of this HROM is to obtain a prediction of the entire mechanical state of the weld after the entire operation. Two tests were done, the first one uses the predicted fields of the 11th pass shown in Table 5.3 to create a HROM.

HROM from plastic strain extrapolation

1. Compute the simulation of the first 11 passes of the simplified multipass test case.
2. Use the first 10 results to forecast the state of the plastic strain for the 11th pass using the following methods:
 - (a) DMD
 - (b) PFS
 - (c) Combination (Kmeans)
 - (d) Combination (Ward)
3. Compute the displacement and stress fields using the inherent strains method.
4. Calculate a POD basis from the displacement and stress predictions.
5. Create the Reduced Integration Domain (RID).
6. Compute a hyper-reduced solution of the 11th pass.
7. Compare the hyper-reduced displacement and stress to the FE simulation of the 11th pass.

The results obtained with the Hyper-reduction method are compared to the FE simulations. Two different HROMs were used, with two sets of parameters. The two parameters are the number of POD modes and the number of extra element layers surrounding the RID. The number of POD modes is chosen by a hyper-parameter μ_{POD} , which is the ratio between the smallest selected singular value and the first singular value of the decomposition. More details on the parameters are given in [45].

The number of extra layers is fixed to 3 extra layers for both HROMs. The number of POD modes is determined by $\mu_{POD} = 10^{-3}$ and $\mu_{POD} = 10^{-5}$. The number of POD modes has a direct influence on the accuracy of the method, as well as the amount of nodes in the RID (as a consequence of using the DEIM [33, 34]). Table 5.5 shows the relative errors between the hyper-reduced solution and the FE simulation. In the case of the prediction done with the PFS method, the number of modes retained when choosing $\mu_{POD} = 10^{-3}$ is very low (only 8 modes), and thus the result is not valid. Setting $\mu_{POD} = 10^{-5}$ improves the prediction of the stress fields with respect to the inherent strains method (see Table 5.3).

TABLE 5.5: Hyper-reduced computations of the 11th pass.

	Error			
	$\mu_{POD} = 10^{-3}$		$\mu_{POD} = 10^{-5}$	
Method	Displacements	Stress	Displacements	Stress
DMD	4.73%	63.31%	3.81%	39.51%
Physical Fields Shift	90.41%	100.03%	0.21%	8.05%
Combination (Kmeans)	6.43%	46.01%	1.03%	20.24%
Combination (Ward)	9.58%	42.40%	0.41%	16.38%

The final test constructs the HROM from the forecasted displacement and stress fields in Table 5.4. The test proceeds as follows:

HROM from consecutive plastic strain extrapolations

1. Compute the simulation of entire simplified multipass test case (20 passes).
2. Use the first 17 results to forecast the state of the plastic strain for the 18th, 19th and 20th passes using the following methods:
 - (a) Combination (Kmeans)
 - (b) Combination (Ward)
3. Compute the displacement and stress fields of the 20th pass using the inherent strains method.
4. Compare the predicted plastic strain, displacement and stress to the FE simulation of the 20th pass.
5. Calculate a POD basis from the displacement and stress predictions.
6. Create the Reduced Integration Domain (RID).
7. Compute a hyper-reduced solution of the 20th pass.
8. Compare the hyper-reduced displacement and stress to the FE simulation of the 20th pass.

The same parameters for the hyper-reduced simulations are used. Table 5.6 summarises the results. In figure 5.6, we can observe a comparison of the final state of stress after the 20th pass. On the left, the RID of a hyper-reduced simulation is presented. From this solution computed on the RID, a solution over the whole domain is obtained by applying the gappy POD method [51]. In the middle, the result of the inherent strain method is shown. On the right, the FE solution of the 20th welding pass. As for the predictions of the 11th pass, the hyper-reduced solutions with $\mu_{POD} = 10^{-5}$ improve the stress estimations of the inherent strains method.

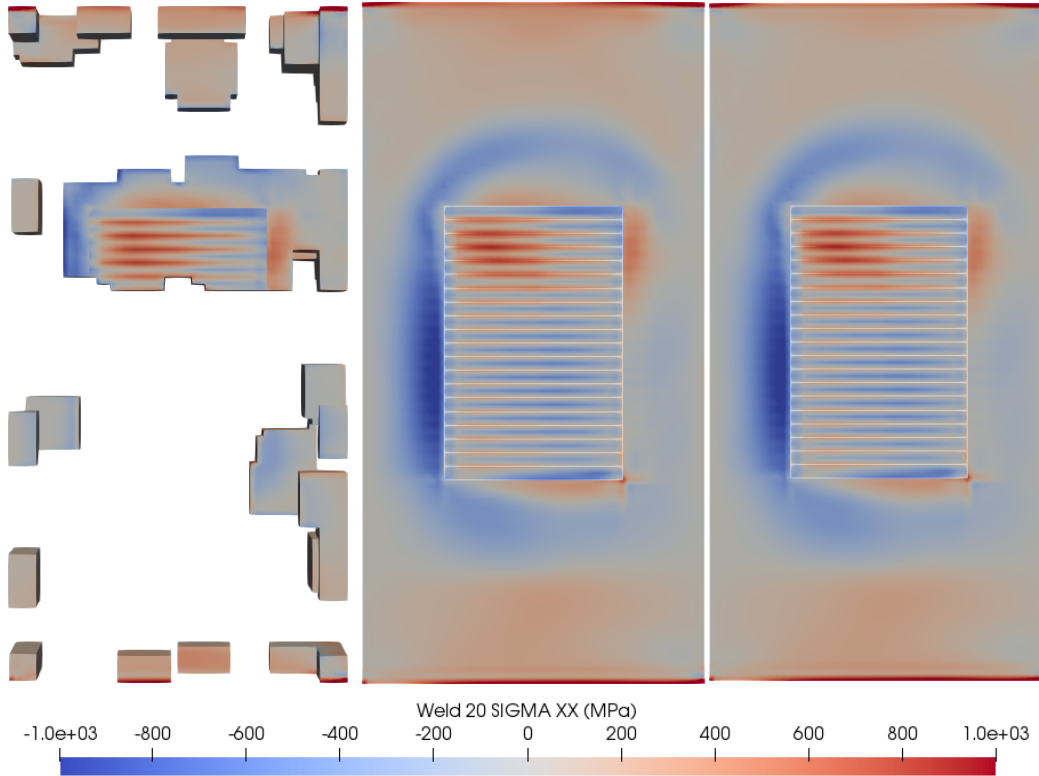


FIGURE 5.6: Comparing the final state of stress (σ_{XX} component) after the 20th pass. The plastic strain predictions were computed using the Ward clustering method. (From left to right: view of the RID of a hyper-reduced simulation, inherent strain, FE simulation).

TABLE 5.6: Hyper-reduced computations of the 11th pass.

	Error			
	$\mu_{POD} = 10^{-3}$		$\mu_{POD} = 10^{-5}$	
Method	Displacements	Stress	Displacements	Stress
Combination (Kmeans)	3.4%	35.06%	1.18%	13.47%
Combination (Ward)	1.18%	64.89%	1.17%	12.5%

5.7.4 Execution time

The main goal of the presented methods is to obtain accurate predictions of future passes using the information of previous ones. The reduction of execution time is nonetheless of great importance. Two different computers were used to compute all the tests presented in the previous subsections. The FE simulations were computed on a scientific cluster using 16 CPUs on two nodes. The other computations were done on a laptop with an i7-6700HQ CPU.

Significant time gains have been observed. On average, the execution time of each of the passes on the scientific cluster (domain decomposition, 2 nodes and 16 CPUs) is 3361 seconds, around 56 minutes. A prediction combining PFS and DMD

computed on laptop takes between 10 and 15 minutes on average, mainly depending on the clustering algorithm used (Kmeans is faster than Ward). For example, to predict the plastic strain of the 20th pass with Kmeans algorithm, 674 seconds of execution time were needed: 11 minutes and 14 seconds.

The inherent strains step is costly when computed on a computer with a single i7 processor and sequential algorithms. Its execution time is, on average, 4109 seconds, around 1 hour and ten minutes. This execution time could be significantly improved if computed on the scientific cluster with parallel algorithms and domain decomposition.

The hyper-reduction parameters determine its duration. A higher number of modes and/or extra element layers lead to a longer execution time. Two configurations were tested, one of them is faster but less accurate, and the other one is more accurate but slower as it takes into account more modes and more elements of the original mesh:

- $\mu_{POD} = 10^{-3}$, 3 extra layers: 658 seconds on average (10 minutes and 58 seconds).
- $\mu_{POD} = 10^{-5}$, 3 extra layers: 1567 seconds on average (26 minutes and 7 seconds).

5.8 Conclusion

Predicting mechanical fields of future passes is an alternative way of accelerating multipass welding simulations. We have presented two extrapolation methods to compute predictions of future passes using the data from the already simulated ones. Each of these methods has advantages and disadvantages that have been pointed out in their respective section. Clustering the regions where they perform better has allowed to improve the accuracy of the predictions.

Two approaches using different mechanical fields have been proposed. On the one hand, if displacement and stress fields are used, the predictions are very accurate with respect to the FE simulations, but we have observed that they might produce results that are not mechanically admissible. On the other hand, if plastic strain fields are used, the predictions are less accurate, but an additional step involving the inherent stress method gives displacement and stress fields from which performing reduced-order models can be computed. Hyper-reduction can then improve the accuracy of the inherent strain results.

There is still some margin to improve the results and the methods. Further and more thorough testing of the presented methods is also suggested for future studies. New prediction methods could also be introduced. For example, a probabilistic approach could be explored. Treating the multipass welding problem as a time series would allow us to use autoregressive models to describe it.

One of the main problems encountered when using these methods is the large size of the files involved. Numerical simulation of welding is not only costly in computation time but also in terms of storage.

The presented results point towards new strategies to accelerate multipass welding simulations involving many passes or to accelerate similar problems such as additive manufacturing simulation.

Chapter 6

Conclusion and perspectives

In the context of the nuclear industry, strict safety standards demand a high level of control of welding maintenance and manufacturing operations. It is of utmost importance to increase the reliability of the procedures and identify defects as early as possible. Currently, most quality controls are performed a posteriori. In the case of non-conformity, the weld has to be repaired, delaying the operation, which has time and financial consequences. Real-time monitoring of welding processes would allow to assess the weld's mechanical state and, thus, optimise the planning of maintenance and repair operations by increasing the procedure's reliability and decreasing the need for inspection. Accurate assessment of a weld's mechanical state in real-time based on the operation's physics is one of EDF's ambitions for the coming years. This PhD work represents the first step in this direction.

The main objective of the PhD was to obtain real-time estimations of temperature and stress fields from partial temperature observations. High-fidelity simulation gives access to unobservable physical quantities that are helpful for post-operation analysis, but it is not fit for real-time applications due to the non-linear multiphysics nature of welding. The increasing availability of data is pushing the boundaries of what can be achieved by integrating classical numerical analysis, reduced-order models and machine learning. We have proposed an offline-online approach that allows us to use rich simulation offline data and integrates observed experimental data in real-time for anomaly detection. The models presented in this document link observed surface temperature measurements obtained with infrared cameras with unobservable mechanical quantities of interest through high-fidelity digital twinning. The offline-online techniques allow achieving the real-time objective without sacrificing accuracy.

We have presented the whole experimental and numerical methodology for monitoring a welding experiment, as well as the construction of the reduced-order models and their use in real-time. Outlooks into the acceleration of challenging multipass welding operations were also explored.

PVR experiments and digital twin

In Chapter 2 we presented the PVR experiment, as well as the experimental and numerical campaigns, which served as a basis for all the developments of the thesis. The PVR hot cracking experiment was selected for the complexity of the physical phenomena and its ability to acquire different types of data. Numerically modelling the thermo-mechanical behaviour of the specimen represents a complicated challenge that includes the calibration of Goldak's equivalent heat source model. Such calibrations have been performed using punctual temperature measurements given by instrumented experiments with thermocouples.

Besides, the instrumentation of the experiment also gave access to surface temperature measurements. The infrared images were used as input data for the model, and comparisons between thermocouple and infrared surface measurements were used to estimate an observation error with the camera. A more extensive calibration phase of the measurement error could be envisioned. Temperatures are deduced from the digital levels of the camera without knowledge of 316L stainless steel's emissivity by a multi-filter approach based on Planck's equation. The temperature values are then projected onto the Finite Element mesh so they can be used with the models.

The high-fidelity simulations of the PVR experiment are produced by solving a weakly coupled thermo-mechanical problem using the Finite Elements Method with EDF's open-source software code `_aster`. It results in a simulation divided into two separate steps. The first step is to solve a non-linear heat problem with a moving heat source. The resulting temperature field is then used as input for a thermo-mechanical problem solved with an elasto-visco-plastic Chaboche behaviour law. A high-fidelity parametric study involving known process parameters and unknown modelling parameters constitutes the basis for the methods introduced in the subsequent chapters. Such a database involves large amounts of data that need to be treated efficiently for real-time applications.

PPCA local surrogate models

The main obstacles to obtaining real-time 3D temperature and stress estimations are the high dimensions of the data and the computation time constraints. The method proposed in Chapter 3 tackles all the dimensional and time-restriction difficulties at the same time by use of the local PPCA models. The reduced bases deal with the large dimension of the simulation data. Furthermore, using local models instead of a single global model helps consider the non-linear parametric variation of the temperature and mechanical fields. This piecewise linear approach is key for real-time efficiency, as it allows to optimise the linear temperature observations of the weld.

The infrared camera is the only source of experimental data in the current work. There are no experimental devices capable of measuring quantities of interest, such as the stress tensor, during the operation, which is an additional significant difficulty. The lack of mechanical data highlights the need for performing thermo-mechanical models, which are the only source of knowledge on the mechanical state of the weld. These models have to accurately describe the physical state of the specimen in order to obtain meaningful estimations from a temperature field. Using multiphysics state vectors has allowed us to estimate fields that are never observed, thanks to the correlations that appear in the reduced bases. Moreover, the multiphysics state vectors include the known and unknown parameters, which allows the estimation of the unknown parameters as well as using the known parameters as an input observation.

Constructing the local bases is as expensive as computing an SVD of the snapshot matrix, so there is no additional cost associated with using PPCA bases instead of the more common PCA or POD bases. However, the multiphysics vector distribution is endowed with a Gaussian prior when using the probabilistic version. This Gaussian prior is particularly useful because it allows solving the inverse problem in a Bayesian way without being confronted with costly sampling methods like MCMC. What is more, an explicit expression of the posterior mean and variance exists as a consequence of the Gaussian modelling. Through algebraic manipulation and the presence of the reduced bases, the posterior mean and variance of the state can be evaluated extremely quickly by only computing low-rank operations with the involved matrices and vectors.

The local models are computed after partitioning the domain in regions according to the heat source position. This physical partition leads to a sequence of models determined by the progression of the torch. New states are estimated by finding the

adequate model for the new infrared image. Two scenarios were considered: one in which the torch position was perfectly known, making the model selection trivial, and one in which there is an uncertainty on the torch position. This second case reflects a situation where no sensor signals the position of the torch or even cases where the different data acquisition devices are not well synchronised. Uncertainty in the knowledge of the heat source position is addressed by a kriging surrogate model that associates the highest temperature with the torch position. This strategy was shown to work for a real PVR experiment where the position of the torch was unknown.

The method presented in Chapter 3 is also well suited to forecast future states. Indeed, considering the parameter estimation of a previous infrared image as noisy knowledge of the parameters, an estimation of a future state is available. However, the forecasted state will have a large confidence region because no data has been observed. Nevertheless, the reduced bases contain enough data related to the physical behaviour of the system that the mean of the posterior forecast is not far from the “true solution” provided by simulation data.

The accuracy of the predicted states and the computation times are encouraging and show potential for the industrialisation of the method. Different extensions for the presented approaches can be envisioned. First of all, by extending the simulation time domain and filming the specimen during the cooling phase, estimation of residual stresses could be possible.

However, obtaining experimental data might be challenging in a non-laboratory environment. For example, using infrared cameras might not be possible because of their physical dimensions. What is more, external factors might compromise the image quality. High-quality image acquisition during an industrial welding application can be challenging because of the reflection from the arc. In the case of PVR experiments, this was overcome by filming the backside of the experiment. In the case of an industrial repair operation on a pipe, this might be impossible to do.

Furthermore, most repair and assembly operations involve a filler metal in the weld, and this approach has not been tested for the more challenging multipass welding. For mono-bead welding, the application of the method should not vary all that much, but new challenges may arise. The models should be constructed with a mesh containing the whole structure, and the bead’s nodes should be activated on the fly.

Another limitation is that for each new type of welding operation, a new physical model has to be created. Indeed, a new geometry or parameter domain needs a new FEM formulation and a subsequent parametric study similar to the one performed in Chapter 2 in order to construct the local reduced models. This issue could be addressed by generating reduced models in the welding torch referential. Assuming that all welding operations are similar locally (in a neighbourhood of the weld and with similar enough curvature), these reduced bases would be useful for a more extensive range of operations that would now differ by the boundary conditions around the torch’s neighbourhood. This is not a flawless solution, as this kind of approach would need a secondary coupled global model in order to predict the state of the rest of the structure, increasing the complexity of the numerical schemes.

Clustering local reduced bases

The local basis approach presented in Chapter 3 relied on the natural choice of the heat source progression as the means of partitioning the simulation snapshots in order to create the local models. Chapter 4 presented an exploratory work in which the partitioning of the parameter domain is performed automatically via clustering. Three proposed metrics are presented with the goal of grouping similar states into clusters to create models that capture the physics associated with the clustered snapshots. These three metrics are the Euclidean distance in the state space, the Projection Error between snapshots and the Grassmann distance between preliminary local reduced bases.

Clustering similar data together helps reduce the variance in the local models but makes the online assignment a more difficult task. In the present case, a new snapshot is given the label of the cluster whose local PPCA model minimises its reconstruction error. This objective-based criterion ensures the optimality of assignment, as the selected model produces the best approximation of the initial snapshot. The assignment in the online stage is done by a neural network classifier trained on the simulation data.

The results of this approach were obtained from noisy simulated data, where the noise amplitude on the temperature field is the same that was estimated in the experimental infrared data. In most cases, it was shown that the estimations were very accurate with respect to the original multiphysics snapshots. The Projection Error clustering, based on Amsallem's work [6], proved to produce the most accurate estimation results in both temperature and principal stress.

However, forecasting future states is difficult with this approach. Contrary to the position-based local models in Chapter 3, there is no easy way to interpret what model should be selected for a future state. A solution to this problem could be to extrapolate the current estimation and create new data from the original input at a later position. This newly generated image would then be given as input to the neural network to indicate what local model to use. It would work in a similar way as a filtering scheme such as Kalman Filters, where a prediction of the future state is done and is later corrected by the observations. In this potential approach, the prediction would be used to find the local model. The main challenge is to define how to generate the new data for future positions and if it can all be done in real-time.

The numerical results presented in Chapter 4 illustrate the relationship between the classification accuracy and a low state estimation error. It was shown that accurate estimations depend on assigning the appropriate local model correctly. A potential improvement in the classification could be achieved by changing the neural network's architecture. Recent work on convolutional neural networks has shown promising results by applying convolution directly to mesh nodes [86], which could be an alternative to the more classic MLP architecture described in Chapter 4. Additionally, based on the notion of classification via logistic regression, other possibilities involve the renumbering of classes such that consecutive classes share similar model properties. In that case, a miss-classification of a snapshot would point to a reduced basis with similar characteristics. Ordering the clusters can be achieved by using the Grassmann distance between the local reduced bases associated with each cluster.

Beyond the accuracy and the computation time of the overall approach, attention should be paid to ensuring that the reconstructed state is physically admissible. Indeed, the worst results showed non-physical solutions that should not be accepted. In order to avoid that, physical constraints should be enforced to ensure, for example, the positivity of a temperature field.

Hyper-reduced models for multipass welding

Most maintenance and manufacturing welding operations involve a large number of welding passes. Numerical simulation of multipass welding is a difficult challenge due to the enormous computation times needed for the entire procedure. In this context, we have proposed two extrapolation techniques as well as different ways of combining them to produce training data for welding beads that have not been simulated yet. After simulating the deposition of the first beads, the mechanical fields are extrapolated to obtain an estimation of the final state of the weld after the deposition of all the beads.

The main goal of those extrapolations is to construct hyper-reduced models that allow producing accurate predictions of the mechanical state of the structure after the operation. The hyper-reduced simulations are meant to improve the prediction of stresses by taking into account the problem's physics. This approach would save time compared to high-fidelity non-linear simulations of every bead without sacrificing accuracy in the predictions.

However, this exploratory work needs further testing in order to validate it. More thorough analysis of the accuracy of the predictions with respect to the FE simulations is mandatory, in particular when extrapolating the results for consecutive beads. Comparing the Grassmann distance between the subspaces spanned by the reduced bases created by FE simulations and by extrapolation could also be useful. This distance would indicate if the hyper-reduced models approximate the FE simulations on similar subspaces when using the extrapolations as training data.

Extensions of these methods could be applied to other thermo-mechanical procedures such as wire-arc additive manufacturing. This additive manufacturing method is similar to welding and could benefit from accelerated simulations for post manufacturing quality assessment.

Industrial perspectives

The industrialisation of the presented methods requires additional work. It has already been stated that most maintenance and manufacturing operations involve a significant number of welding passes. However, the real-time approaches developed during this PhD have not been tested for a welding experiment with filler metal. The next step toward industrialisation would be an experimental proof of concept in which a single bead is deposited. Then, subsequent tests on simple multipass cases should be performed.

A possible test configuration would be a simplified groove weld. This test presents two major challenges, the first one being the already mentioned filler metal. The other one is related to the data acquisition with an infrared camera in a lab environment, which would need improvement in order to avoid reflections from the arc.

Overcoming those challenges would be a significant step toward real-time control of industrial operations that can be done in a welding workshop.

Appendix A

Appendix A: Projection error algorithm

Abstract:

This appendix will come back to the Projection Error clustering presented in Chapter 4. In particular, the effect of the original initialisation step of the algorithm proposed by Amsallem in [6] is shown.

Résumé :

Cet annexe revient sur la méthode de clustering par Erreur de Projection présentée dans le Chapitre 4. En particulier, on regardera l'effet de l'étape d'initialisation originale de l'algorithme proposé par Amsallem dans [6].

The hierarchical divisive algorithm presented in Algorithms 4.2 and 4.3 starts with a single clustering that contains all the snapshots. The new clusters are created by extracting the snapshots that have the largest reconstruction error with respect to a representative snapshot. In Amsallem's article [6], the representative of this initial snapshot is chosen as the snapshot with the maximum L^2 norm, whilst our version uses the median L^2 norm.

The results that we obtained with Amsallem's initialization lead to a first cluster where there is no concentration of snapshots whatsoever, as shown in Figure A.1. This is not necessarily wrong, but it is the only cluster where this happens. All the other ones behave similarly as the cluster shown in Figure 4.5 obtained with the present method: clusters are centered around certain positions.

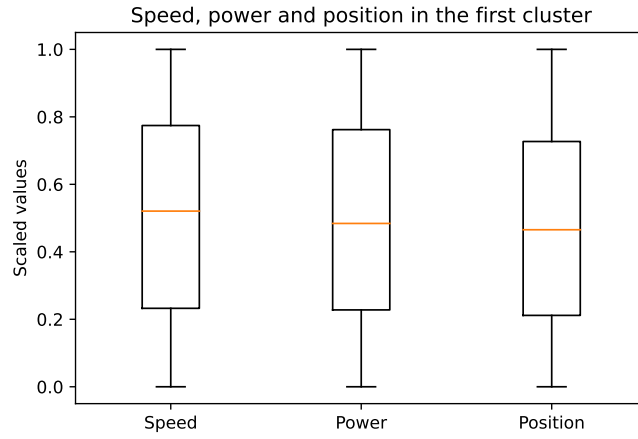


FIGURE A.1: Distribution of speed, power and position for the first cluster when initialising with the maximum L^2 norm.

In order to verify that this cluster is abnormal, we calculated the reconstruction error of all snapshots of a cluster with respect to the basis associated to the cluster, and then we calculated the average reconstruction error per cluster Δ_i :

$$\bar{\Delta}_i = \frac{1}{\#C_i} \sum_{s \in C_i} \delta_R(s, \Phi^i), \quad \forall i \in \llbracket 1, \dots, K \rrbracket \quad (\text{A.1})$$

where the PPCA reconstruction error δ_R is presented in Equation (4.22) and $\#$ is the cardinal operator.

Whilst all clusters have an average reconstruction error between 0 and 2 %, the first cluster has an average reconstruction error of 15.31%, more than seven times than the next cluster.

This is not the case when using the median L^2 norm as the first vector, which is the example shown in Figure 4.5. Here, the first cluster is also the largest by a great margin: the first cluster has 1877 snapshots, more than three times the size of the next biggest cluster (554 snapshots). However, the average PPCA reconstruction error in this cluster is 1.87%.

We have not found yet an explanation on why there is such a difference between the two initialisation steps, but the correction that we implemented definitely seems to solve the issue.

Bibliography

- [1] Mickaël Abbas. *La méthode des éléments finis isoparamétriques*. https://www.code-aster.org/V2/doc/default/fr/man_r/r3/r3.01.00.pdf. 2013.
- [2] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. “Riemannian geometry of Grassmann manifolds with a view on algorithmic computation”. In: *Acta Applicandae Mathematica* 80.2 (2004), pp. 199–220.
- [3] Grégoire Allaire. *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*. OUP Oxford, 2007.
- [4] Grégoire Allaire et al. *Numerical linear algebra*. Vol. 55. Springer, 2008.
- [5] David Amsallem and Charbel Farhat. “Interpolation method for adapting reduced-order models and application to aeroelasticity”. In: *AIAA journal* 46.7 (2008), pp. 1803–1813.
- [6] David Amsallem and Bernard Haasdonk. “PEBL-ROM: Projection-error based local reduced-order models”. In: *Advanced Modeling and Simulation in Engineering Sciences* 3.1 (2016), pp. 1–25.
- [7] David Amsallem, Matthew J Zahr, and Charbel Farhat. “Nonlinear model order reduction based on local reduced-order bases”. In: *International Journal for Numerical Methods in Engineering* 92.10 (2012), pp. 891–916.
- [8] David Amsallem, Matthew J Zahr, and Kyle Washabaugh. “Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction”. In: *Advances in Computational Mathematics* 41.5 (2015), pp. 1187–1230.
- [9] David Amsallem et al. “A method for interpolating on manifolds structural dynamics reduced-order models”. In: *International journal for numerical methods in engineering* 80.9 (2009), pp. 1241–1258.
- [10] David Arthur and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.
- [11] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.
- [12] N. Aubry et al. “The Dynamics Of Coherent Structures In The Wall Region Of A Turbulent Boundary-Layer”. In: *Journal of Fluid Mechanics* 192 (1988), pp. 115–173.
- [13] François Auger et al. “Industrial applications of the Kalman filter: A review”. In: *IEEE Transactions on Industrial Electronics* 60.12 (2013), pp. 5458–5471.
- [14] Ivo Babuska et al. *The finite element method and its reliability*. Oxford university press, 2001.

- [15] NR Baddoo. “Stainless steel in construction: A review of research, applications, challenges and opportunities”. In: *Journal of constructional steel research* 64.11 (2008), pp. 1199–1206.
- [16] Joan Baiges, Ramon Codina, and Sergio Idelsohn. “Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations”. In: *International Journal for Numerical Methods in Fluids* 72.12 (2013), pp. 1219–1243. DOI: <https://doi.org/10.1002/fld.3777>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3777>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3777>.
- [17] Raj Baldev et al. “Development of stainless steels in nuclear industry: With emphasis on sodium cooled fast spectrum reactors history, technology and foresight”. In: *Advanced Materials Research*. Vol. 794. Trans Tech Publ. 2013, pp. 3–25.
- [18] Maxime Barrault et al. “An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations”. In: *Comptes Rendus Mathématique* 339.9 (2004), pp. 667–672.
- [19] Alexander T Basilevsky. *Statistical factor analysis and related methods: theory and applications*. Vol. 418. John Wiley & Sons, 2009.
- [20] Michaël Baudin et al. “Title: Open TURNS: An industrial software for uncertainty quantification in simulation”. In: *Handbook of uncertainty quantification*. Ed. by D. Higdon R. Ghanem and H. Owhadi. Springer, 2017. URL: <https://hal.archives-ouvertes.fr/hal-01107849>.
- [21] Jacques Besson et al. *Mécanique non linéaire des matériaux*. Hermès Science Publ., 2001.
- [22] Christopher M Bishop. “Latent variable models”. In: *Learning in graphical models*. Springer, 1998, pp. 371–403.
- [23] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [24] Diana A. Bistrián and Ionel M. Navon. “The method of dynamic mode decomposition in shallow water and a swirling flow problem”. In: *International Journal for Numerical Methods in Fluids* 83.1 (2016), pp. 73–89. DOI: [10.1002/fld.4257](https://doi.org/10.1002/fld.4257). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4257>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4257>.
- [25] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.
- [26] Géraud Blatman. “Chaos polynomial creux et adaptatif pour la propagation d’incertitudes et l’analyse de sensibilité”. Theses. Université Blaise Pascal - Clermont-Ferrand II, Oct. 2009. URL: <https://tel.archives-ouvertes.fr/tel-00440197>.
- [27] Régis Blondeau, ed. *Métallurgie et mécanique du soudage*. 2001.
- [28] Théo Boutin. “Detection de dérives en temps réel lors de la mise en oeuvre de procédés de soudage par analyse expérimentale et apprentissage automatique”. PhD thesis. Université de Montpellier, 2020-2023.

- [29] Théo Boutin, Sami Hilal, and Pablo Pereira Álvarez. *Approches expérimentales pour l'observation cartographique de la diffusion de la température: calibration d'une caméra infrarouge*. Unpublished internal EDF R&D report. 2020.
- [30] Diego Canales et al. "Vademecum-based GFEM (V-GFEM): optimal enrichment for transient problems". In: *International Journal for Numerical Methods in Engineering* 108.9 (2016), pp. 971–989.
- [31] Kevin Carlberg et al. "The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows". In: *Journal of Computational Physics* 242 (2013), pp. 623–647.
- [32] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. "A comparative study of efficient initialization methods for the k-means clustering algorithm". In: *Expert systems with applications* 40.1 (2013), pp. 200–210.
- [33] Saifon Chaturantabut and Danny C Sorensen. "Discrete empirical interpolation for nonlinear model reduction". In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE. 2009, pp. 4316–4321.
- [34] Saifon Chaturantabut and Danny C Sorensen. "Nonlinear model reduction via discrete empirical interpolation". In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.
- [35] Siddhartha Chib and Edward Greenberg. "Understanding the metropolis-hastings algorithm". In: *The american statistician* 49.4 (1995), pp. 327–335.
- [36] Francisco Chinesta, Pierre Ladeveze, and Elías Cueto. "A short review on model order reduction based on proper generalized decomposition". In: *Archives of Computational Methods in Engineering* 18.4 (2011), p. 395.
- [37] Alejandro Cosimo, Alberto Cardona, and Sergio Idelsohn. "Global-local HROM for non-linear thermal problems with irreversible changes of material states". In: *Comptes Rendus Mécanique* 346.7 (2018). Model reduction, data-based and advanced discretization in computational mechanics / Réduction de modèles, données et techniques de discrétisation avancées en mécanique numérique, pp. 539–555. ISSN: 1631-0721. DOI: <https://doi.org/10.1016/j.crme.2018.04.002>. URL: <https://www.sciencedirect.com/science/article/pii/S1631072118300718>.
- [38] Alejandro Cosimo, Alberto Cardona, and Sergio Idelsohn. "Improving the k-compressibility of hyper reduced order models with moving sources: applications to welding and phase change problems". In: *Computer Methods in Applied Mechanics and Engineering* 274 (2014), pp. 237–263.
- [39] Katalin Csilléry et al. "Approximate Bayesian computation (ABC) in practice". In: *Trends in ecology & evolution* 25.7 (2010), pp. 410–418.
- [40] Tiangang Cui, Youssef M Marzouk, and Karen E Willcox. "Data-driven model reduction for the Bayesian solution of inverse problems". In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 966–990.

- [41] Thomas Daniel et al. “Model order reduction assisted by deep neural networks (ROM-net)”. In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (2020), pp. 1–27.
- [42] Masoumeh Dashti and Andrew M. Stuart. “The Bayesian Approach to Inverse Problems”. In: *Handbook of Uncertainty Quantification*. Cham: Springer International Publishing, 2017, pp. 311–428. ISBN: 978-3-319-12385-1. DOI: [10.1007/978-3-319-12385-1_7](https://doi.org/10.1007/978-3-319-12385-1_7). URL: https://doi.org/10.1007/978-3-319-12385-1_7.
- [43] Lionel Depradeux. “Simulation numérique du soudage-acier 316L: validation sur cas tests de complexité croissante”. PhD thesis. Villeurbanne, INSA, 2004.
- [44] Markus Dihlmann, Martin Drohmann, and Bernard Haasdonk. “Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning”. In: *Proc. of ADMOS 2011* (2011), p. 64.
- [45] Tuan Dinh Trong. “Modèles hyper-réduits pour la simulation simplifiée du soudage en substitut de calcul hors d’atteinte”. PhD thesis. Mines ParisTech, 2018.
- [46] Martin Drohmann, Bernard Haasdonk, and Mario Ohlberger. “Adaptive Reduced Basis Methods for Nonlinear Convection–Diffusion Equations”. In: *Finite Volumes for Complex Applications VI Problems & Perspectives*. Ed. by Jaroslav Fořt et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 369–377. ISBN: 978-3-642-20671-9.
- [47] Electricité de France. *Finite element code_aster, Structures and Thermo-mechanics Analysis for Studies and Research*. Open source on www.code-aster.org. 1989–2022.
- [48] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*. Vol. 159. Springer, 2004.
- [49] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.
- [50] Geir Evensen. “The ensemble Kalman filter: Theoretical formulation and practical implementation”. In: *Ocean dynamics* 53.4 (2003), pp. 343–367.
- [51] Richard Everson and Lawrence Sirovich. “Karhunen–Loeve procedure for gappy data”. In: *JOSA A* 12.8 (1995), pp. 1657–1664.
- [52] Kai-Tai Fang, Runze Li, and Agus Sudjianto. *Design and modeling for computer experiments*. CRC press, 2005.
- [53] Charbel Farhat, Todd Chapman, and Philip Avery. “Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models”. In: *International journal for numerical methods in engineering* 102.5 (2015), pp. 1077–1110.
- [54] Carolin Fink, Manuela Zinke, and Daniel Keil. “Evaluation of Hot Cracking Susceptibility of Nickel- Based Alloys by The Pvr Test”. In: *Welding in the World* 56 (2012), pp. 37–43.

- [55] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. Vol. 20. Siam, 2007.
- [56] M.W Gardner and S.R Dorling. “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences”. In: *Atmospheric Environment* 32.14 (1998), pp. 2627–2636. ISSN: 1352-2310. DOI: [https://doi.org/10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0). URL: <https://www.sciencedirect.com/science/article/pii/S1352231097004470>.
- [57] Marc G. Genton. “Classes of Kernels for Machine Learning: A Statistics Perspective”. In: *J. Mach. Learn. Res.* 2 (Mar. 2002), pp. 299–312. ISSN: 1532-4435.
- [58] Roger Ghanem, David Higdon, Houman Owhadi, et al. *Handbook of uncertainty quantification*. Vol. 6. Springer, 2017.
- [59] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical approximation of hyperbolic systems of conservation laws*. Vol. 118. Springer Science & Business Media, 2013.
- [60] John A Goldak and Mehdi Akhlaghi. *Computational welding mechanics*. Springer Science & Business Media, 2005.
- [61] John Goldak, Aditya Chakravarti, and Malcolm Bibby. “A new finite element model for welding heat sources”. In: *Metallurgical transactions B* 15.2 (1984), pp. 299–305.
- [62] Bernard Haasdonk, Markus Dihlmann, and Mario Ohlberger. “A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space”. In: *Mathematical and Computer Modelling of Dynamical Systems* 17.4 (2011), pp. 423–442. DOI: [10.1080/13873954.2011.547674](https://doi.org/10.1080/13873954.2011.547674). eprint: <https://doi.org/10.1080/13873954.2011.547674>. URL: <https://doi.org/10.1080/13873954.2011.547674>.
- [63] Bernard Haasdonk and Mario Ohlberger. “Reduced basis method for finite volume approximations of parametrized linear evolution equations”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 42.2 (2008), pp. 277–302.
- [64] Jacques Hadamard. “Sur les problèmes aux dérivées partielles et leur signification physique”. In: *Princeton university bulletin* (1902), pp. 49–52.
- [65] Martin T Hagan, Howard B Demuth, and Mark Beale. *Neural network design*. PWS Publishing Co., 1997.
- [66] Janelle Katherine Hammond et al. “PBDW: A non-intrusive Reduced Basis Data Assimilation method and its application to an urban dispersion modeling framework”. In: *Applied Mathematical Modelling* 76 (2019), pp. 1–25.
- [67] Per Christian Hansen. “The L-curve and its use in the numerical treatment of inverse problems”. In: (1999).
- [68] J Hassan, A Majid Awan, and A Jalil. “Welding defect detection and classification using geometric features”. In: *2012 10th International Conference on Frontiers of Information Technology*. IEEE. 2012, pp. 139–144.
- [69] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

- [70] Harold V Henderson and Shayle R Searle. “On deriving the inverse of a sum of matrices”. In: *Siam Review* 23.1 (1981), pp. 53–60.
- [71] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*. Vol. 590. Springer, 2016.
- [72] H.D. Hibbitt and P.V. Marcal. “A numerical, thermo-mechanical model for the welding and subsequent loading of a fabricated structure”. In: *Computers and Structures* (1973), pp. 1145–1174. DOI: [10.1016/0045-7949\(73\)90043-6](https://doi.org/10.1016/0045-7949(73)90043-6).
- [73] Dave Higdon et al. “Computer model calibration using high-dimensional output”. In: *Journal of the American Statistical Association* 103.482 (2008), pp. 570–583.
- [74] Sami Hilal. “Simulation thermo-mécanique du procédé de fabrication additive par dépôt-fil WAAM”. PhD thesis. Mines ParisTech, 2022.
- [75] Michael R Hill and Drew V Nelson. “The inherent strain method for residual stress determination and its application to a long welded joint”. In: *ASME-PUBLICATIONS-PVP* 318 (1995), pp. 343–352.
- [76] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.
- [77] Tristan S Hunnewell et al. “Total hemispherical emissivity of SS 316L with simulated very high temperature reactor surface conditions”. In: *Nuclear Technology* 198.3 (2017), pp. 293–305.
- [78] T. Inoue. “3 - Thermal-metallurgical-mechanical interactions during welding”. In: *Processes and Mechanisms of Welding Residual Stress and Distortion*. Ed. by Zhili Feng. Woodhead Publishing Series in Welding and Other Joining Technologies. Woodhead Publishing, 2005, pp. 99–125. ISBN: 978-1-85573-771-6. DOI: <https://doi.org/10.1533/9781845690939.1.99>. URL: <https://www.sciencedirect.com/science/article/pii/B9781855737716500033>.
- [79] Rudolph Emil Kalman. “A new approach to linear filtering and prediction problems”. In: (1960).
- [80] Thomas Kannengiesser and Thomas Boellinghaus. “Hot cracking tests: an overview of present technologies and applications”. In: *Welding in the World* 58.3 (2014), pp. 397–421.
- [81] L. Karlsson. *Thermal stresses in welding*. R.B. Hetnarski (Ed.), Thermal stresses I, Elsevier Science Publishers. 1986.
- [82] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.
- [83] Pierre Kerfriden et al. “Bridging proper orthogonal decomposition methods and augmented Newton–Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems”. In: *Computer Methods in Applied Mechanics and Engineering* 200.5-8 (2011), pp. 850–866.

- [84] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [85] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).
- [86] Vasilis Krokos et al. *A Graph-based probabilistic geometric deep learning framework with online physics-based corrections to predict the criticality of defects in porous materials*. Submitted for publication to IJNME. 2022. DOI: [10.48550/ARXIV.2205.06562](https://doi.org/10.48550/ARXIV.2205.06562).
- [87] Inna Krykova. *Evaluating of path-dependent securities with low discrepancy methods*. Master of Science Thesis, Worcester Polytechnic Institute. 2004.
- [88] P Ladevèze and P GERMAIN. “La méthode à grand incrément de temps pour l’analyse de structures à comportement non linéaire décrit par variables internes”. In: *Comptes rendus de l’Académie des sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l’univers, Sciences de la Terre* 309.11 (1989), pp. 1095–1099.
- [89] Kody Law, Andrew Stuart, and Kostas Zygalakis. “Data assimilation”. In: *Cham, Switzerland: Springer* 214 (2015).
- [90] Minh Le. “Approches expérimentale et numérique de la fissuration à chaud dans les soudures en acier inoxydable”. Thèse de doctorat dirigée par Pilvin, Philippe Sciences de l’ingénieur Lorient 2014. PhD thesis. 2014, 1 vol. (178 p.) URL: <http://www.theses.fr/2014LORIS326>.
- [91] Jean Lemaitre et al. *Mécanique des matériaux solides-3e éd.* Dunod, 2020.
- [92] Randall J LeVeque et al. *Finite volume methods for hyperbolic problems*. Vol. 31. Cambridge university press, 2002.
- [93] YC Liang et al. “Proper orthogonal decomposition and its applications—Part I: Theory”. In: *Journal of Sound and vibration* 252.3 (2002), pp. 527–544.
- [94] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. “The global k-means clustering algorithm”. In: *Pattern recognition* 36.2 (2003), pp. 451–461.
- [95] Lars-Erik Lindgren. *Computational welding mechanics*. Elsevier, 2014.
- [96] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: [1711.05101](https://arxiv.org/abs/1711.05101) [cs.LG].
- [97] Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2017. arXiv: [1608.03983](https://arxiv.org/abs/1608.03983) [cs.LG].
- [98] Ye Lu et al. “Adaptive hyper reduction for additive manufacturing thermal fluid analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113312. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2020.113312>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782520304977>.
- [99] John Leask Lumley. “The structure of inhomogeneous turbulent flows”. In: *Atmospheric turbulence and radio wave propagation* (1967).

- [100] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [101] Yvon Maday and Einar M Rønquist. “A reduced-basis element method”. In: *Comptes Rendus Mathématique* 335.2 (2002), pp. 195–200.
- [102] Yvon Maday and Tommaso Taddei. “Adaptive PBDW approach to state estimation: noisy observations; user-defined update spaces”. In: *SIAM Journal on Scientific Computing* 41.4 (2019), B669–B693.
- [103] Yvon Maday et al. “A general multipurpose interpolation procedure: the magic points”. In: *Communications on Pure & Applied Analysis* 8.1 (2009), p. 383.
- [104] Yvon Maday et al. “A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics”. In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 933–965.
- [105] Yvon Maday et al. “The generalized Empirical Interpolation Method: stability theory on Hilbert spaces with an application to the Stokes equation”. In: *Computer Methods in Applied Mechanics and Engineering* 287 (2015), pp. 310–334.
- [106] Jean-Michel Marin et al. “Approximate Bayesian computational methods”. In: *Statistics and Computing* 22.6 (2012), pp. 1167–1180.
- [107] Amandine Marrel et al. “An efficient methodology for modeling complex computer codes with Gaussian processes”. In: *Computational Statistics & Data Analysis* 52.10 (2008), pp. 4731–4744.
- [108] Youssef M Marzouk and Habib N Najm. “Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems”. In: *Journal of Computational Physics* 228.6 (2009), pp. 1862–1902.
- [109] Youssef M Marzouk, Habib N Najm, and Larry A Rahn. “Stochastic spectral methods for efficient Bayesian solution of inverse problems”. In: *Journal of Computational Physics* 224.2 (2007), pp. 560–586.
- [110] Tomas Mikolov and Geoffrey Zweig. “Context dependent recurrent neural network language model”. In: *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2012, pp. 234–239.
- [111] J Mirapeix et al. “Real-time arc-welding defect detection and classification with principal component analysis and artificial neural networks”. In: *NDT & e International* 40.4 (2007), pp. 315–323.
- [112] Rolando Mosquera et al. “POD basis interpolation via Inverse Distance Weighting on Grassmann manifolds”. In: *Discrete & Continuous Dynamical Systems-S* 12.6 (2019), p. 1743.
- [113] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL: probml.ai.

- [114] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL: probml.ai.
- [115] Ionel M. Navon. “Data Assimilation for Numerical Weather Prediction: A Review”. In: *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 21–65. ISBN: 978-3-540-71056-1. DOI: [10.1007/978-3-540-71056-1_2](https://doi.org/10.1007/978-3-540-71056-1_2). URL: https://doi.org/10.1007/978-3-540-71056-1_2.
- [116] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [117] Hae-Sang Park and Chi-Hyuck Jun. “A simple and fast algorithm for K-medoids clustering”. In: *Expert systems with applications* 36.2 (2009), pp. 3336–3341.
- [118] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [119] José M Pena, Jose Antonio Lozano, and Pedro Larranaga. “An empirical comparison of four initialization methods for the k-means algorithm”. In: *Pattern recognition letters* 20.10 (1999), pp. 1027–1040.
- [120] Pablo Pereira Álvarez and David Ryckelynck. *Extrapolation of welding simulations via dynamic modes for the estimation of POD modes*. Unpublished internal EDF R&D and Mines ParisTech report. 2019.
- [121] Pablo Pereira Álvarez et al. “A shift technique for multipass welding simulation”. In: *Mathematical Modelling of Weld Phenomena*. Ed. by C. Sommitsch, N. Enzinger, and P. Mayr. Vol. 12. The 12th International Seminar Numerical Analysis of Weldability ; Conference date: 23-09-2018 Through 26-09-2018. Verlag der Technischen Universität Graz, 2019. DOI: [10.3217/978-3-85125-615-4](https://doi.org/10.3217/978-3-85125-615-4). URL: <https://www.tugraz.at/events/seggau/seminar-information/home/>.
- [122] Pablo Pereira Álvarez et al. “Real-Time Data Assimilation in Welding Operations Using Thermal Imaging and Accelerated High-Fidelity Digital Twinning”. In: *Mathematics* 9.18 (2021). ISSN: 2227-7390. DOI: [10.3390/math9182263](https://doi.org/10.3390/math9182263). URL: <https://www.mdpi.com/2227-7390/9/18/2263>.
- [123] Christophe PrudHomme et al. “Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods”. In: *Journal of Fluids Engineering* 124.1 (Nov. 2001), pp. 70–80. DOI: [10.1115/1.1448332](https://doi.org/10.1115/1.1448332). URL: <https://hal.archives-ouvertes.fr/hal-00798326>.
- [124] G Rabensteiner, J Tosch, and H Schabereiter. “Hot cracking problems in different fully austenitic weld metals.” In: *WELDING J.* 62.1 (1983), p. 21.

- [125] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- [126] CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006, p. 248.
- [127] Waseem Rawat and Zenghui Wang. “Deep convolutional neural networks for image classification: A comprehensive review”. In: *Neural computation* 29.9 (2017), pp. 2352–2449.
- [128] Michał Jerzy Rewieński. “A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems”. PhD thesis. Massachusetts Institute of Technology, Department of Electrical Engineering . . . , 2003.
- [129] Maria Isabel Ribeiro. “Kalman and extended kalman filters: Concept, derivation and properties”. In: *Institute for Systems and Robotics* 43 (2004), p. 46.
- [130] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*. Vol. 2. Springer, 1999.
- [131] I.B.C.M. Rocha, P. Kerfriden, and F.P. van der Meer. “On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning”. In: *Journal of Computational Physics: X* 9 (2021), p. 100083. ISSN: 2590-0552. DOI: <https://doi.org/10.1016/j.jcpx.2020.100083>. URL: <https://www.sciencedirect.com/science/article/pii/S2590055220300354>.
- [132] Peter J. Rousseeuw. “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7). URL: <https://www.sciencedirect.com/science/article/pii/0377042787901257>.
- [133] Peter J. Rousseeuw and Leonard Kaufman. “Clustering by means of medoids”. In: *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*. Vol. 31. 1987.
- [134] Sam Roweis. “EM algorithms for PCA and SPCA”. In: *Advances in neural information processing systems* (1998), pp. 626–632.
- [135] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations”. In: *Archives of Computational Methods in Engineering* 15.3 (2007), p. 1.
- [136] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations”. In: *Archives of Computational Methods in Engineering* 15.3 (2008), pp. 229–275.

- [137] Gianluigi Rozza and Karen Veroy. “On the stability of the reduced basis method for Stokes equations in parametrized domains”. In: *Computer methods in applied mechanics and engineering* 196.7 (2007), pp. 1244–1260.
- [138] Paul-Baptiste Rubio, Ludovic Chamoin, and François Louf. “Real-time data assimilation and control on mechanical systems under uncertainties”. In: *Advanced Modeling and Simulation in Engineering Sciences* 8.1 (2021), pp. 1–25.
- [139] Axel Ruhe. “Rational Krylov sequence methods for eigenvalue computation”. In: *Linear Algebra and its Applications* 58 (1984), pp. 391–405.
- [140] David Ryckelynck. “A priori hyperreduction method: an adaptive approach”. In: *Journal of computational physics* 202.1 (2005), pp. 346–366.
- [141] David Ryckelynck. “Hyper-reduction of mechanical models involving internal variables”. In: *International Journal for Numerical Methods in Engineering* 77.1 (2009), pp. 75–89.
- [142] David Ryckelynck, Thibault Goessel, and Franck Nguyen. “Mechanical dissimilarity of defects in welded joints via Grassmann manifold and machine learning”. en. In: *Comptes Rendus. Mécanique* 348.10-11 (2020), pp. 911–935. DOI: [10.5802/crmeca.51](https://doi.org/10.5802/crmeca.51).
- [143] David Ryckelynck et al. “On the a priori model reduction: overview and recent developments”. In: *Archives of Computational methods in Engineering* 13.1 (2006), pp. 91–128.
- [144] Ekkehard W Sachs and Stefan Volkwein. “POD-Galerkin approximations in PDE-constrained optimization”. In: *GAMM-Mitteilungen* 33.2 (2010), pp. 194–208.
- [145] Andrea Saltelli. “Making best use of model evaluations to compute sensitivity indices”. In: *Computer Physics Communications* 145.2 (2002), pp. 280–297. ISSN: 0010-4655. DOI: [https://doi.org/10.1016/S0010-4655\(02\)00280-1](https://doi.org/10.1016/S0010-4655(02)00280-1). URL: <https://www.sciencedirect.com/science/article/pii/S0010465502002801>.
- [146] Peter J Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.
- [147] Erich Schubert and Peter J. Rousseeuw. “Fast and eager k-medoids clustering: O(k) runtime improvement of the PAM, CLARA, and CLARANS algorithms”. In: *Information Systems* 101 (2021), p. 101804. ISSN: 0306-4379. DOI: <https://doi.org/10.1016/j.is.2021.101804>. URL: <https://www.sciencedirect.com/science/article/pii/S0306437921000557>.
- [148] Erich Schubert and Peter J. Rousseeuw. “Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms”. In: *Similarity Search and Applications*. Ed. by Giuseppe Amato et al. Cham: Springer International Publishing, 2019, pp. 171–187. ISBN: 978-3-030-32047-8.
- [149] Erich Schubert et al. “DBSCAN revisited, revisited: why and how you should (still) use DBSCAN”. In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), pp. 1–21.

- [150] Lawrence Sirovich. “Turbulence and the dynamics of coherent structures. I. Coherent structures”. In: *Quarterly of applied mathematics* 45.3 (1987), pp. 561–571.
- [151] Lawrence Sirovich and Michael Kirby. “Low-dimensional procedure for the characterization of human faces”. In: *Josa a* 4.3 (1987), pp. 519–524.
- [152] MC Smith et al. “Accurate prediction of residual stress in stainless steel welds”. In: *Computational Materials Science* 54 (2012), pp. 312–328.
- [153] Christian Soize and R. Ghanem. “Physical systems with random uncertainties: Chaos representations with arbitrary probability measure”. In: *SIAM Journal on Scientific Computing* 26.2 (2004), pp. 395–410. DOI: [10.1137/S1064827503424505](https://doi.org/10.1137/S1064827503424505). URL: <https://hal-upec-upem.archives-ouvertes.fr/hal-00686211>.
- [154] Donald F Specht et al. “A general regression neural network”. In: *IEEE transactions on neural networks* 2.6 (1991), pp. 568–576.
- [155] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.
- [156] P. Tekriwal and J. Mazumder. “Finite element analysis of three-dimensional transient heat transfer in GMA welding”. In: *Weld J* 67 (1988), pp. 150–156.
- [157] The HDF Group. *Hierarchical Data Format, version 5*. <https://www.hdfgroup.org/HDF5/>. 1997-2022.
- [158] Michael E Tipping and Christopher M Bishop. “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.
- [159] G. Tran Van et al. “Effect of Boron Content on Hot Ductility and Hot Cracking Susceptibility in 316L Austenitic Stainless Steel for Welding Components”. In: *Journal of Materials Engineering and Performance* 27.10 (Oct. 2018), pp. 5114–5123. ISSN: 1544-1024. DOI: [10.1007/s11665-018-3640-z](https://doi.org/10.1007/s11665-018-3640-z). URL: <https://doi.org/10.1007/s11665-018-3640-z>.
- [160] Giai Tran Van. “Determination of a liquation hot cracking criterion as a function of boron content and its location for 316L austenitic stainless steel”. Thesis. Université de Bretagne Sud, Dec. 2018. URL: <https://tel.archives-ouvertes.fr/tel-02134428>.
- [161] John W Tukey et al. *Exploratory data analysis*. Vol. 2. Reading, MA, 1977.
- [162] Yukio Ueda, Keiji Fukuda, and Masayuki Tanigawa. “New measuring method of three dimensional residual stresses based on theory of inherent strain (welding mechanics, strength & design)”. In: *Transactions of JWRI* 8.2 (1979), pp. 249–256.
- [163] Yukio UEDA et al. “A New Measuring Method of Residual Stresses with the Aid of Finite Element Method and Reliability of Estimated Values”. In: *Transactions of JWRI* 4.2 (1975), pp. 123–131.

- [164] Daniele Venturi and George Em Karniadakis. “Gappy data and reconstruction procedures for flow past a cylinder”. In: *Journal of Fluid Mechanics* 519 (2004), pp. 315–336.
- [165] Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002.
- [166] Stefan Volkwein. “Model reduction using proper orthogonal decomposition”. In: *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz*. see <http://www.uni-graz.at/imawww/volkwein/POD.pdf> (2011).
- [167] Baicun Wang et al. “Intelligent welding system technologies: State-of-the-art review and perspectives”. In: *Journal of Manufacturing Systems* 56 (2020), pp. 373–391.
- [168] Kyle Washabaugh et al. “Nonlinear model reduction for CFD problems using local reduced-order bases”. In: *42nd AIAA Fluid Dynamics Conference and Exhibit*. 2012, p. 2686.
- [169] Greg Welch, Gary Bishop, et al. “An introduction to the Kalman filter”. In: (1995).
- [170] B. P. Welford. “Note on a Method for Calculating Corrected Sums of Squares and Products”. In: *Technometrics* 4.3 (1962), pp. 419–420. DOI: [10.1080/00401706.1962.10490022](https://doi.org/10.1080/00401706.1962.10490022). eprint: <https://www.tandfonline.com/doi/pdf/10.1080/00401706.1962.10490022>. URL: <https://www.tandfonline.com/doi/abs/10.1080/00401706.1962.10490022>.
- [171] Karen Willcox. “Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition”. In: *Computers & fluids* 35.2 (2006), pp. 208–226.
- [172] Karen Willcox and Jaime Peraire. “Balanced model reduction via the proper orthogonal decomposition”. In: *AIAA journal* 40.11 (2002), pp. 2323–2330.
- [173] PJ Withers et al. “Recent advances in residual stress measurement”. In: *International Journal of Pressure Vessels and Piping* 85.3 (2008), pp. 118–127.
- [174] Zonghan Wu et al. “A comprehensive survey on graph neural networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.
- [175] Ke Ye and Lek-Heng Lim. “Schubert varieties and distances between subspaces of different dimensions”. In: *SIAM Journal on Matrix Analysis and Applications* 37.3 (2016), pp. 1176–1197.
- [176] Yancheng Zhang, Alain Combescure, and Anthony Gravouil. “Efficient hyper reduced-order model (HROM) for parametric studies of the 3D thermo-elasto-plastic calculation”. In: *Finite Elements in Analysis and Design* 102 (2015), pp. 37–51.
- [177] X.K. Zhu and Y.J. Chao. “Effects of temperature-dependent material properties on welding simulation”. In: *Computers & Structures* 80.11 (2002), pp. 967–976. ISSN: 0045-7949. DOI: [https://doi.org/10.1016/S0045-7949\(02\)00040-8](https://doi.org/10.1016/S0045-7949(02)00040-8). URL: <http://www.sciencedirect.com/science/article/pii/S0045794902000408>.



RÉSUMÉ

Nous proposons le développement d'une stratégie d'assimilation de données offline/online en temps réel pour l'estimation de champs physiques et paramètres inconnus basée sur des métamodèles probabilistes locaux et thermographie infrarouge. Les apports récents des algorithmes d'apprentissage automatique et des méthodes de réduction de modèle permettent d'obtenir des modèles phénoménologiques exploitables en temps réel afin d'évaluer un risque de dérive lors du soudage. Ces métamodèles sont construits à partir de simulations haute-fidélité par éléments finis faites offline. Online, les prévisions en temps réel seront réalisées à l'aide de métamodèles de champs multiphysiques et d'observations surfaciques de température par caméra infrarouge.

MOTS CLÉS

Réduction de modèle, Assimilation de données, Temps réel, Soudage.

ABSTRACT

We propose developing a real-time offline/online data assimilation strategy for the estimation of physical fields and unknown parameters based on local probabilistic surrogate models and infrared thermography. The recent contributions of machine learning algorithms and model order reduction methods make it possible to obtain phenomenological models that can be used in real-time to evaluate the risk of drift during welding. These surrogate models are built from high-fidelity finite element simulations performed offline. Online, real-time predictions will be made using multiphysics surrogate models and surface temperature observations obtained with an infrared camera.

KEYWORDS

Model Order Reduction, Data Assimilation, Real-time, Welding.

