



HAL
open science

Physics-augmented neural networks for constitutive modeling

Antoine Benady

► **To cite this version:**

Antoine Benady. Physics-augmented neural networks for constitutive modeling. Solid mechanics [physics.class-ph]. Université Paris-Saclay, 2024. English. NNT : 2024UPAST089 . tel-04887187

HAL Id: tel-04887187

<https://theses.hal.science/tel-04887187v1>

Submitted on 14 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Physics-augmented neural networks for constitutive modeling

*Réseaux de neurones augmentés par la physique pour
l'apprentissage de lois de comportement*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°579 : Sciences Mécaniques et Énergétiques, Matériaux et
Géosciences (SMEMaG)

Spécialité de doctorat : Mécanique des solides et des structures

Graduate School : Sciences de l'ingénierie et des systèmes

Référent : ENS Paris-Saclay

Thèse préparée dans l'unité de recherche **LMPS** - Laboratoire de Mécanique Paris-Saclay
(Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS),
sous la direction de **Ludovic CHAMOIN**, Professeur des Universités et le co-encadrement
d'**Emmanuel BARANGER**, directeur de recherche CNRS.

Thèse soutenue à Paris-Saclay, le 18 juillet 2024, par

Antoine BENADY

Composition du jury

Membres du jury avec voix délibérative

| | |
|---|--------------|
| Pierre KERFRIDEN Professeur des Universités, Mines Paris | Président |
| Elias CUETO Professeur des Universités, Universidad de Zaragoza. | Rapporteur |
| Eric FLORENTIN Professeur des Universités, Institut National des Sciences Appliquées Centre Val de Loire | Rapporteur |
| Emmanuelle ABISSET-CHAVANNE Professeure des Universités, Ecole Nationale Supérieure d'Arts et Métiers | Examinatrice |

Titre : Réseaux de neurones augmentés par la physique pour l'apprentissage de lois de comportement. **Mots clés :** Erreur en Relation de Comportement modifiée, Réseau de neurones augmentés par la physique, Loi de comportement non-linéaire, Thermodynamique, Filtre de Kalman.

Résumé :

Le contrôle de santé des structures demeure une préoccupation essentielle en ingénierie, étant donné les impératifs de sécurité et de durabilité. Une pratique récente consiste à mettre en place un dialogue entre une structure physique et son jumeau numérique, afin de prédire l'état de santé et de limiter l'endommagement.

Cette thèse s'intéresse à la question fondamentale de la construction automatique d'un modèle de comportement matériau, nécessaire à la prédiction fiable de l'état de la structure. Les lois de comportement matériau sont représentées par des réseaux de neurones contraints à respecter les principes de la thermodynamique, via l'apprentissage de potentiels convexes intervenant dans le cadre des Matériaux Standards Généralisés. L'entraînement des réseaux de neurones est réalisé de

manière non supervisée par la minimisation de l'erreur en relation de comportement modifiée (mCRE). La fonctionnelle mCRE offre un sens physique riche grâce à l'indicateur d'erreur de modèle ainsi qu'à la possibilité de garantir le respect des connaissances fiables. Une procédure de calibration automatique des hyperparamètres est développée pour réduire la sensibilité aux choix de l'utilisateur. La méthode développée dans cette thèse est testée sur différents types de comportements non-linéaires (hyperélasticité, élastoplasticité, viscoplasticité).

Enfin, une procédure d'assimilation de données, fondée sur les filtres de Kalman, est développée pour prédire de manière séquentielle l'état de la structure, dans le cas de comportement matériau dépendant de l'histoire.

Title : Physics-augmented neural networks for constitutive modeling

Keywords : Modified Constitutive Relation Error, Physics-augmented neural networks, Nonlinear constitutive law, Thermodynamics, Kalman Filter.

Abstract : Structural health monitoring remains a crucial concern in engineering, given the imperatives of safety and durability. A recent practice involves establishing a dialogue between a physical structure and its digital twin to predict the health status and limit damage.

This thesis focuses on the fundamental question of automatically constructing a material constitutive model, necessary for reliably predicting the structure's state. Material constitutive laws are represented by neural networks constrained to adhere to the principles of thermodynamics, through the learning of convex potentials within the framework of Generalized Standard Materials. The training of neural net-

works is performed in an unsupervised manner by minimizing the modified constitutive relation error (mCRE). The mCRE functional provides a rich physical sense due to the model error indicator and the ability to guarantee compliance with reliable knowledge. An automatic hyperparameter calibration procedure is developed to reduce sensitivity to user choices. The method developed in this thesis is tested on various types of nonlinear behaviors (hyperelasticity, elastoplasticity, viscoplasticity).

Finally, a data assimilation procedure based on Kalman filters is developed to sequentially predict the structure's state, in the case of material behavior dependent on history.

À Lucienne Benady

Acknowledgements

L'aboutissement de cette thèse marque une étape significative dans mon parcours académique et personnel, et ce n'aurait pas été possible sans le soutien, l'encadrement et les encouragements de nombreuses personnes. Je tiens à exprimer ma plus profonde gratitude à celles et ceux qui, par leur présence et leurs conseils, ont contribué à la réalisation de ce travail.

Je tiens tout d'abord à exprimer ma profonde gratitude aux membres de mon jury de thèse : Je remercie tout particulièrement Monsieur Pierre Kerfriden d'avoir accepté de présider mon jury de thèse, et pour la qualité des échanges que nous avons pu avoir. J'adresse également mes sincères remerciements à Messieurs Elias Cueto et Eric Florentin, rapporteurs de cette thèse, pour leur lecture attentive, leurs retours constructifs et les réflexions qu'ils m'ont inspirées. Enfin, je souhaite remercier Madame Emmanuelle Abisset-Chavanne pour sa participation active et ses observations pertinentes lors de la soutenance.

Je tiens ensuite à exprimer toute ma gratitude à mes directeurs de thèse pour cette aventure formidable. Ludovic Chamoin, pour la confiance que tu m'as accordée en me proposant ce sujet passionnant. Ton accompagnement constant, combiné à la liberté de penser que tu m'as laissée pour explorer et me développer en tant que chercheur, a été une source inestimable de motivation. Grâce à toi, j'ai eu l'opportunité de participer à diverses conférences à l'étranger, ce qui a enrichi ma formation et élargi mes horizons scientifiques. Ton expertise, ta vision et ton humanité m'ont permis d'aborder ce travail avec optimisme et sérénité. Emmanuel Baranger, pour ton écoute attentive et ton soutien indéfectible tout au long de cette thèse, ton exigence dans la bienveillance, ton esprit critique si nécessaire dans l'océan de publications en IA, ainsi que, évidemment, pour les BBQ annuels. J'ai énormément grandi à vos côtés et j'aurai à cœur de transmettre à mon tour ce que vous m'avez transmis.

Ces travaux de thèse sont aussi le fruit de nombreuses discussions avec les permanents que j'ai côtoyés pendant ces 3 dernières années : P. Ladevèze, O. Allix, PAB, F. Louf, PAG, F. Feyel, F. Daghia, M. Poncelet, L. Fribourg, F. Amlani, A. Barbarulo, H. Ben Dhia, R. Desmorat, F. Hild, F. Gatti, G. Puel et A.-L. Hamon.

Cette thèse a surtout été une aventure collective à côté de tous les copains. MERCI à tous. Sofiane et Valentin, pour votre présence sans laquelle je ne serais peut-être pas

allé au bout de cette thèse ainsi que pour notre tentative de groupe de musique. Pierre, mon compagnon d'IRTG, sans qui certaines semaines auraient été vraiment interminables. Mahmoud, merci d'être Mahmoud. Marie, Breno, Maaarc, Héloïse, les dromadaires du midi. Sahar et Daniel, merci pour notre collaboration du projet ERC qui a été une vraie source de motivation. Clément, ton arrivée au laboratoire a marqué un véritable nouveau souffle dans ma thèse et a été déterminante dans la dernière ligne droite. Merci à l'équipe du CDS : Victor, Quentin, Karim, Pierre-Éliot, Floriane, Alexandre, Flavien. Sylvain, merci pour toutes nos discussions passionnantes. Merci Matthieu D. pour nos précieuses discussions sur la mCRE (et sur le chef). Xavier et Ahmed, merci pour votre bonne humeur et nos discussions sur le foot. Merci aussi à tous ceux de l'équipe des foots du lundi (sauf Donald ;). Merci Stiven pour nos discussions sur l'IA et notre aventure singapourienne. Martin P., merci pour tes suggestions cinématographiques et photographiques. Merci Patrick pour tes conseils de vie inestimables. Merci Edgar et Vikram pour vos travaux de stage qui m'ont beaucoup aidé. Merci à l'équipe de Modeo. Merci aux copains de l'IRTG Fynn, Julian, Marius, Jan, Hendrik, Swami, Ammar. Merci également à tous les autres pour les bons moments passés au laboratoire : Gwendal, Alexis, Daniel, Matthieu, Afsal, Fatima, Stiven, Louis, Corentin, Omar, Rami, Rodrigo, Hilario, Claire, Julien, Yann, Bastien, Matteo, Max, Filipe, Clément, Camille, Xiaofei, Rodolfo, Elio, Lucas, Elvis !

J'aimerais également prendre le temps de remercier l'équipe du centre de gestion et surtout Charlène et Lydia pour leur travail indispensable au fonctionnement du laboratoire.

Merci à Larissa, Cimino et Remo pour votre soutien pendant cette thèse.

Enfin, je tiens à exprimer ma plus profonde gratitude à ma famille, dont le soutien inébranlable, l'amour inconditionnel et la patience infinie m'ont permis de mener à bien ce projet. Merci à mes parents, Aurélie et Philippe, de mon frère Charles, de mes grands-parents, oncles et tantes, cousins et cousines.

Pour ceux que j'aurais malencontreusement oubliés, sachez que les réclamations ne changeront rien puisqu'il m'est impossible de modifier ce manuscrit. En revanche, je saurai me rattraper avec des chocolats.

Contents

| | |
|--|-----------|
| Acknowledgements | iii |
| Contents | v |
| Introduction | 1 |
| 1 Literature review on the modified Constitutive Relation Error framework for inverse problems, and Deep Learning for constitutive modeling | 7 |
| 1 The mCRE framework | 8 |
| 1.1 The CRE concept for verification of finite element method results .. | 8 |
| 1.2 The mCRE for inverse problem | 10 |
| 1.3 The mCRE minimization for linear elasticity | 11 |
| 1.4 Previous works on mCRE | 15 |
| 1.5 The mCRE framework for nonlinear behavior | 16 |
| 2 Coupling Deep Learning and constitutive modeling | 20 |
| 2.1 Introduction to Deep Learning | 20 |
| 2.2 Coupling techniques between physical knowledge and neural networks | 27 |
| 2.3 Neural networks and constitutive modeling | 30 |
| 2 Learning nonlinear state laws with neural networks trained in the mCRE framework | 39 |
| 1 Problem definition | 40 |
| 2 A minimization procedure suited for nonlinear elasticity | 41 |
| 2.1 Step 1 | 42 |
| 2.2 Step 2 | 44 |
| 3 Automatic rules for hyperparameters tuning | 45 |
| 3.1 Reducing the sensitivity to initialization: physics-guided initialization with <i>a priori</i> knowledge | 46 |
| 3.2 Reducing the sensitivity to weighting between losses: automatic tuning of α | 46 |
| 3.3 Reducing the sensitivity to learning rate: automatic tuning of learning rate | 47 |
| 3.4 Reducing the sensitivity to the number of epochs: definition of a physics-based stop criterion | 49 |
| 3.5 Extension to the case of multiple loadings in the database | 50 |

| | | |
|----------|--|------------|
| 4 | Practical implementation | 50 |
| 5 | Results | 51 |
| 5.1 | Application on a nonlinear case in small deformation | 51 |
| 5.2 | Application on a hyperelastic case: Mooney-Rivlin model | 54 |
| 5.3 | Robustness on hyperparameters choice | 57 |
| 3 | Learning nonlinear history-dependent behaviors with neural networks trained in the mCRE framework | 63 |
| 1 | Problem definition | 64 |
| 1.1 | A minimization procedure suited for history-dependent behavior | 65 |
| 2 | Training NN with the mCRE framework | 69 |
| 2.1 | Constitutive model described by a thermodynamically-consistent neural network | 70 |
| 2.2 | Hyperparameters automatic tuning strategy | 70 |
| 3 | Results | 72 |
| 3.1 | A simple 1D example to identify an isotropic hardening modulus | 73 |
| 3.2 | Rate-independent test case: learning a nonlinear isotropic hardening law | 76 |
| 3.3 | Rate-dependent test case: learning a dissipation pseudo-potential | 83 |
| 4 | On the need for an adequate parameterization of the neural networks | 91 |
| 1 | The problem of local minima with NN-mCRE | 92 |
| 1.1 | The specificities of the NN-mCRE with initialization | 92 |
| 1.2 | An example of initialization bias | 92 |
| 1.3 | Possibilities to get rid of initialization bias | 94 |
| 2 | An adequate parameterization of neural networks for constitutive model | 95 |
| 2.1 | A one-layer modified ICNN | 96 |
| 2.2 | Expressivity of one-layer modified ICNN | 99 |
| 2.3 | Interpretable extrapolation offered by one-layer modified ICNN | 99 |
| 2.4 | Discussion on the use of second-order optimization method | 101 |
| 3 | Results with one-layer modified ICNN | 102 |
| 3.1 | Example with a nonlinear elastic behavior | 102 |
| 3.2 | Example of interpretable extrapolation in elastoplasticity | 103 |
| 5 | A Modified Dual Kalman Filtering approach for sequential data-assimilation involving nonlinear evolution laws | 109 |
| 1 | From Linear Kalman Filter to Modified Dual Kalman Filter | 112 |
| 1.1 | Linear Kalman Filter (LKF) | 113 |
| 1.2 | Nonlinear extensions of KF: focus on the Scaled Spherical Simplex Filter (S3F) | 114 |
| 1.3 | Kalman Filter for parameter identification | 117 |
| 2 | Extension of MDKF to address evolution laws | 118 |
| 3 | Results | 120 |

| | | |
|----------|--|------------|
| 4 | Toward a coupling between MDKF and model bias correction with neural networks | 120 |
| | Conclusion | 125 |
| | Appendices | 129 |
| A | Comparison between NN-EUCLID and NN-mCRE | 131 |
| 1 | EUCLID framework in brief | 131 |
| 2 | Comparison of NN-mCRE and NN-EUCLID | 133 |
| B | Learning a hyperelastic behavior with the NN-EUCLID method applied on experimental data | 135 |
| C | Repository with the Python tutorials | 141 |
| D | Publications and communications | 143 |
| E | Extended abstract in french | 147 |
| | Bibliography | 151 |

Introduction

Controlling structural integrity to prevent failure caused by manufacturing defects, impacts, excessive loading, fatigue, or other unexpected events is a critical engineering concern. Many structures, such as wind turbines, aircraft, bridges, satellites, and vehicles, may be susceptible to potential degradations. In this context, it is crucial to implement techniques that can detect damage early on and track their growth to ensure the integrity of structures throughout their lifespan (Farrar & Worden, 2007). Ensuring the durability of structures not only addresses security challenges but also has profound environmental implications. Structures with long lifetimes reduce the need for frequent replacement, thereby reducing the consumption of raw materials and energy required for new construction. The Intergovernmental Panel on Climate Change (IPCC) highlights the importance of sustainable infrastructure development in mitigating the effects of climate change and emphasizes the role of durability in achieving these goals (Intergovernmental Panel on Climate Change, 2019). By focusing on improving the durability and resilience of structures with Structural Health Monitoring (SHM), researchers and engineers can make a significant contribution to global sustainability and environmental protection efforts.

SHM has evolved significantly over the years. A recently published literature review (Wang & Ke, 2024) identifies four progressive levels of SHM: damage detection, damage localization, damage quantification, and damage prognosis. The first level, damage detection, involves identifying the presence of damage within a structure (Rytter, 1993). Following detection, the second level, damage localization, seeks to determine the precise location of damage (Doebeling et al., 1996). The third level, damage quantification, assesses the severity of the damage (Farrar & Worden, 2007). Finally, the fourth level, damage prognosis, predicts the future state of structural health and any potential impact on the structure performance (Si et al., 2011). The present thesis, part of an academic project, aims to develop the numerical tools for the future of model-based damage prognosis. Today, achieving efficient SHM reaching damage prognosis is still an open challenge which now seems possible because of a combination of different factors: high-resolution sensors, accurate physics-based models, and fast simulation and data assimilation tools (Chamoin, 2021).

The ingredients for modern SHM: high-resolution sensors, accurate physics-based models, and fast simulation and data assimilation tools.

The first component of modern SHM is the use of sensing techniques that allow in-situ measurements with high spatial resolution, enabling rapid and continuous damage detec-

tion. Modern sensors are progressively replacing traditional point sensors (strain gauges or displacement transducers) as they provide much richer experimental information. These powerful sensing techniques include image-based approaches such as Digital Image Correlation (DIC) (Sutton et al., 1983; Hild & Roux, 2006), vibration-based sensing (Doebbling et al., 1998; Chesné & Deraemaeker, 2013), embedded micro-sensor arrays using electrically conductive carbon nanotubes (Thostenson & Chou, 2006; Guzman De Vitoria et al., 2011) or optical fibers (Froggatt & Moore, 1998; Glisic & Inaudi, 2007). The present manuscript focuses on an embedded network of optical fibers based on Rayleigh backscattering, see (Chamoin et al., 2022) for a review. In this technology, local modifications of the fiber properties lead to variations in the magnitude of the backscattered multi-frequency signal (caused by the interaction of light with natural density fluctuations within the fibers), which is analyzed by Fast Fourier Transform (Soller et al., 2005; Hénault et al., 2012), measuring both static and dynamic strain fields (Schwartz, 2008; Rausch & Mäder, 2010).

The second factor is the rich knowledge history of physics-based constitutive models (Lemaitre & Chaboche, 1990). They are of practical importance for predicting material behavior inside mechanical structures, from damage initiation to failure. Complex constitutive models are now used on Finite Element virtual testing, reducing the need for long and costly experimental campaigns (Abisset, 2012; Liu & Zheng, 2010). A large set of simulation models is then available, ranging from macro-level to micro-level damage descriptions with increasing complexity and CPU cost (hierarchical modeling) (Fish, 2011). The scale of modeling used in this manuscript is based on the thermodynamics of continuous media with a description based on internal variables (Chaboche, 1988, 1993; Ladevèze et al., 1991).

Another critical aspect is the need for real-time simulation and data assimilation. Applied mathematics, computational mechanics, and computer science have made significant contributions, particularly in the development of reduced-order modeling (ROM) techniques (Barrault et al., 2004; Chinesta et al., 2014; Quarteroni et al., 2011). These techniques have become a cornerstone in addressing the computational challenges inherent to large-dimensional engineering problems that persist despite continuous improvements in computing resources. These approaches significantly reduce CPU costs and memory requirements without sacrificing solution accuracy (Chinesta et al., 2011). This manuscript does not focus on the use of ROM techniques and the speed-up of the developed numerical methods will be discussed in the conclusion as a research perspective.

The DREAM-ON project: building Dynamic Data Driven Application Systems (DDDAS) for SHM.

The modern sensing technologies, the rich history of physics-based modeling, as well as the efficient ROM techniques open up the possibility to design new processes in which simulations and measurements dynamically and seamlessly exchange information to control systems. This is the motivation for a paradigm called Dynamic Data Driven Applications

Systems (DDDAS) (Blasch et al., 2022). The DDDAS concept is one of the most challenging applications of Simulation-Based Engineering Sciences (SBES) (Oden et al., 2006). It involves the use of systems that provide relevant data in (near) real-time to computational models of the evolution of physical phenomena of interest in order to predict and control outputs and meet a set of objectives. The main idea of DDDAS is therefore to establish a feedback loop between the real system and its numerical simulator (twin), whose objectives are twofold:

- to continuously predict the evolution of relevant physical phenomena and adjust the system accordingly (e.g. adjust the orientation of a wind turbine blade);
- to dynamically update the computational model by assimilating in real-time some in-situ measurements for effective diagnosis and prognosis.

The DDDAS concept, by simultaneously incorporating theoretical and experimental information, increases the reliability of simulation models by benefiting from the recent trend in SHM. It relies on hybrid approaches (Maday et al., 2015; Peherstorfer & Willcox, 2015b; Chinesta et al., 2019) to take the best of both model-based and data-based approaches. Model-based approaches benefit from high interpretability and physical consistency: they have low variance and high bias. Data-based approaches are good at fitting observations at the cost of poor generalization and significant sensitivity to measurement noise: they have high variance and low bias. The use of hybrid methods can dramatically improve the capabilities for decision-making and control of the evolutionary system. Establishing such a synergistic and continuous dialog between simulations and data is gradually becoming an attractive key challenge in the industry and is expected to be an important future technology (Darema, 2015).

The ERC project DREAM-ON (Chamoin, 2021), of which this work is a part, aims to address the key numerical challenges needed to achieve accurate and efficient SHM in the DDDAS framework. Figure 1 illustrates the general idea of this project. In-situ measurements (from optic fiber sensors) are sequentially assimilated to create an updated model, referred to as a "hybrid twin" of "digital twin instance" (McClellan et al., 2022). This updated model is then used for safety control of the structural health.

The challenges addressed in this thesis: predicting the material behavior for the construction of a digital twin.

This thesis focuses on the construction of the hybrid twin. The key challenge addressed in this thesis is to accurately predict the structural state with a focus on the modeling of the material behavior. Constitutive modeling requires a deep understanding of the material behavior, and although there is a large literature on physics-based material models, an accurate description of the structural state throughout the life cycle is complicated to achieve. This challenge raises 2 questions. The first one, which is the main focus of this

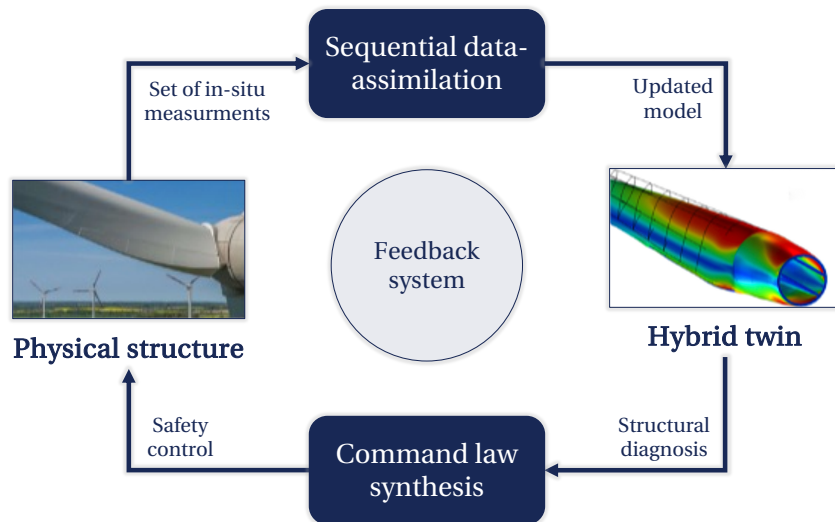


Figure 1 • General description of the ERC project DREAM-ON

thesis, aims to automatically define and adapt the constitutive model all along the life cycle of the structure. This question is referred to as the **model bias correction** or model enrichment, and is performed with historical data. The second one aims to use on-the-fly measurements with the updated constitutive model to sequentially predict the structural state. This question, referred to as **sequential data-assimilation**, is only a secondary goal of this thesis which was treated in collaboration with another PhD student of the DREAM-ON project. Figure 2 summarizes these two questions for the construction of the hybrid twin.

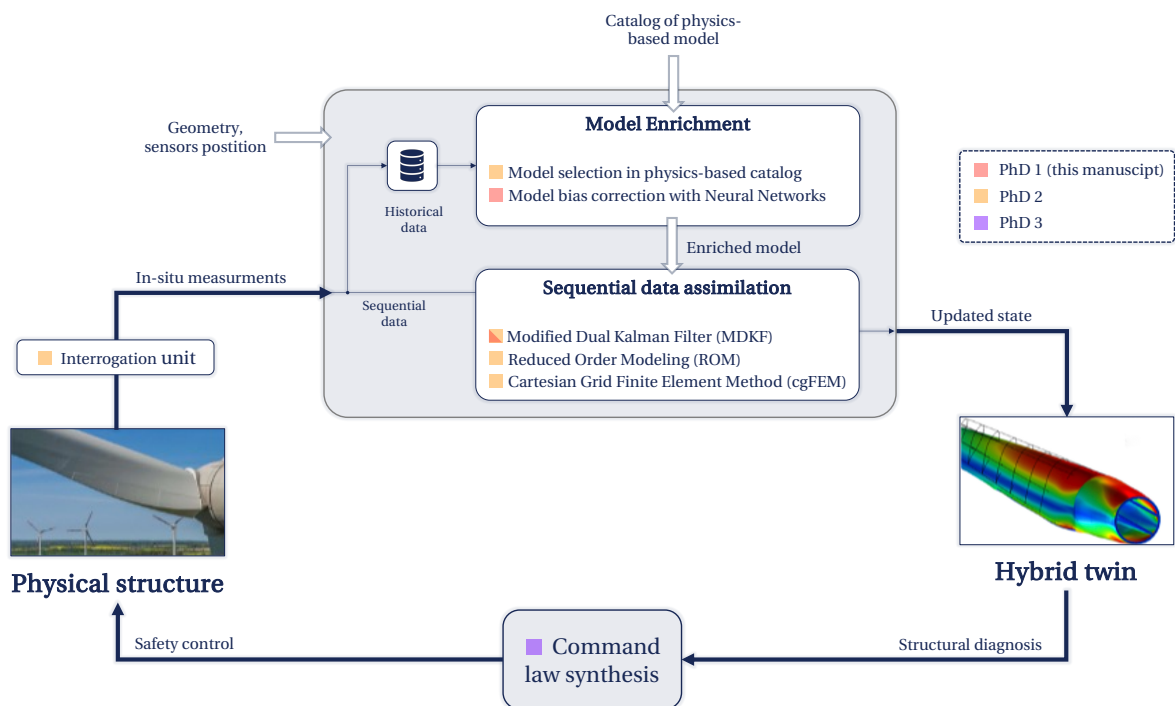


Figure 2 • Detailed description of the ERC project DREAM-ON

To generate an accurate constitutive model from data, the observations can be used to

select the best model from a predefined catalog of physics-based models (Flaschel et al., 2021). However, even the best model from a predefined catalog is likely to have a model bias. One way to avoid model bias is not to postulate any model form. This can be done by replacing the model with tools that can represent functions such as random forests (Breiman, 2001), gradient boosting machines (Friedman, 2001), support vector machines (Cortes & Vapnik, 1995), or neural networks (Goodfellow et al., 2016). The works of this manuscript are based on the latter tool and benefit from the extensive trend of coupling physical knowledge with neural networks (Raissi et al., 2019). With the pioneering works in the 1990s (Ghaboussi et al., 1999), an active research direction is now the use of neural networks to represent the constitutive model (Dornheim et al., 2024). The use of neural networks for constitutive modeling raises several issues.

A primary concern is how to take advantage of the rich history of modeling, in which we can distinguish two levels of knowledge. On the one hand, modeling frameworks have been developed to satisfy thermodynamic principles (Halphen & Nguyen, 1975; Öttinger & Grmela, 1997). These modeling frameworks are fundamental to avoid writing constitutive models that do not obey the laws of thermodynamics. Ideally, a neural network representing a material behavior should be able to fit into one of these modeling frameworks. To ensure compliance with thermodynamic principles, the constitutive neural networks used in this thesis are constrained to formulate models within the Generalized Standard Material framework (Halphen & Nguyen, 1975), with the learning of thermodynamic potentials. On the other hand, within these thermodynamic frameworks, researchers have developed numerous models to best represent material behavior (Lemaitre & Chaboche, 1990): learning a constitutive model from scratch with data, without using existing models, is probably not appropriate (need for large amount of data, complexity of optimization, energy efficiency of the learning process, etc.). It is therefore natural to wonder how to take advantage of these two levels of knowledge when using neural networks to describe a constitutive law.

A second issue concerns the training of neural networks. Learning a constitutive model - relating strain to stress - is not an easy task, since measuring stress is not feasible in practice. As there are no labeled measurements of the input and output, training a constitutive neural network with measurable data is an unsupervised learning task. However, the question of identifying model parameters (not described by neural networks) with the same type of data is not new. This task, referred to as the inverse problem (Bonnet & Constantinescu, 2005), has been widely studied and numerous identification methods already exist, which can be based on least squares minimization (Grédiac, 2004), stochastic approaches (Kaipio & Somersalo, 2007; Tarantola, 2005; Rosic et al., 2013), or minimization of a cost function with a strong physical sense (Claire et al., 2004; Andrieux et al., 1999; Ladevèze et al., 1994). Thus, the unsupervised training of neural networks can naturally benefit from the literature on inverse problems in solid mechanics. Among the identification methods, the modified Constitutive Relation Error (mCRE) framework (Ladevèze et al., 1994) offers very attractive properties in the context of the ERC project DREAM-ON. First, this framework is suited to deal with the available data, which are partial (as opposed to full-field measurements) (Waeytens et al., 2016), noisy (Feissel & Allix, 2007), potentially corrupted (Allix et al., 2005)

and with incomplete boundary conditions (Diaz et al., 2015). Second, the critical context of SHM requires meaningful metrics of prediction reliability. The mCRE function provides a rich physical sense through its modeling error term, available for each prediction, which can localize the region of the structure where the model bias is important (Bui & Constantinescu, 2000; Deraemaeker et al., 2004).

📖 **Goal of this thesis regarding model bias correction**

Regarding the model bias correction goal, the question addressed is the following: How to train neural networks to represent thermodynamic potential in an unsupervised manner with the mCRE framework? This question implies the following questions:

- How to perform the unsupervised neural network training? How to minimize the mCRE?
- How to ensure compliance with thermodynamics?
- How to measure the reliability of the learned constitutive model?

Organization of the manuscript.

The manuscript is structured as follows:

- a bibliographic review is performed in Chapter 1. It focuses on the modified Constitutive Relation Error framework and the coupling techniques between physical knowledge and neural networks, including neural networks for constitutive modeling.
- Chapters 2 and 3 focus on the methodological aspect of training neural networks in the mCRE framework. Chapter 2 focuses on behaviors that are history-independent (nonlinear elasticity) whereas Chapter 3 treats history-dependent behavior (elastoplasticity and viscoplasticity).
- Chapter 4 emphasizes the need for an adequate parameterization of the neural network to achieve universal approximation, allow to leverage of the knowledge of existing models, and improve the training efficiency.
- Finally, Chapter 5 proposes an mCRE-based sequential data assimilation framework suited to deal with history-dependent behavior in the context of SHM with model bias correction. The bibliography relevant to sequential data-assimilation will be introduced in this Chapter.

1

Literature review on the modified Constitutive Relation Error framework for inverse problems, and Deep Learning for constitutive modeling

Contents

| | | |
|----------|---|-----------|
| 1 | The mCRE framework | 8 |
| 1.1 | The CRE concept for verification of finite element method results | 8 |
| 1.2 | The mCRE for inverse problem | 10 |
| 1.3 | The mCRE minimization for linear elasticity | 11 |
| 1.3.1 | Step 1: computation of admissible fields minimizing the mCRE (for fixed parameters) | 11 |
| 1.3.2 | Step 2: updating model parameters with a gradient descent step | 14 |
| 1.4 | Previous works on mCRE | 15 |
| 1.5 | The mCRE framework for nonlinear behavior | 16 |
| 1.5.1 | Thermodynamics framework | 16 |
| 1.5.2 | mCRE formulation for nonlinear behavior | 18 |
| 2 | Coupling Deep Learning and constitutive modeling | 20 |
| 2.1 | Introduction to Deep Learning | 20 |
| 2.1.1 | The multilayer perceptron | 22 |
| 2.1.2 | Other network architectures | 24 |
| 2.1.3 | Gradient-based optimization for neural network training | 25 |
| 2.2 | Coupling techniques between physical knowledge and neural networks | 27 |

| | | |
|-------|---|----|
| 2.2.1 | Physics-guided loss function | 27 |
| 2.2.2 | Physics-guided Architecture | 29 |
| 2.2.3 | Physics-guided initialization | 29 |
| 2.3 | Neural networks and constitutive modeling | 30 |

The introduction of this manuscript has highlighted the importance of accurate constitutive modeling in the context of SHM. Despite the current availability of a wide variety of physical models, model bias will always remain. If the structure of a model is assumed to be too far from reality, it will be impossible to obtain relevant predictions, even with a relevant parameter identification where the model parameters best fit the measured data. One way to free oneself from the a priori writing of a model is to replace the physical model with a neural network. As already mentioned, the use of neural networks has to be done carefully, e.g. not to violate the physics. In this context, training a neural network to represent a constitutive law is simply the next step at the intersection of two different research directions: inverse problems for constitutive modeling and physics-enhanced Deep Learning. On the one hand, training a neural network in an unsupervised context is nothing more than an inverse problem, arguably more difficult due to the significant number of parameters. As mentioned before, an interesting idea is to treat this inverse problem with the mCRE framework, which offers numerous appealing properties to achieve SHM in the DDDAS paradigm. Therefore, the first part of this bibliographic chapter is dedicated to a literature review on the mCRE error. First, the basics of the mCRE error and its minimization are recalled, and then special attention is paid to the formulation of the mCRE error in the context of nonlinear behavior law, specifically formulated with the generalized standard materials framework.

On the other hand, the use of neural networks raises specific questions related to the coupling between this tool of artificial intelligence and physics. The second part of this bibliographic chapter is dedicated to coupling techniques between Deep Learning and physical knowledge. First, the basics of Deep Learning are recalled, then the ways to combine physical knowledge and Deep Learning are classified into three main families of techniques, and finally special attention is paid to the recent use of neural networks in learning constitutive laws.

1 The modified Constitutive Relation Error (mCRE) framework

1.1 The CRE concept for verification of finite element method results

The CRE concept was introduced in the 1970s in the context of finite element (FE) verification (*i.e.*, a posteriori error estimation and mesh adaptation) to define a bound on the FE discretization error. Pioneering ideas are detailed in a series of papers by Ladevèze and co-workers (Ladevèze, 1975; Ladevèze & Leguillon, 1983; Ladevèze & Rougeot, 1997;

Destuynder & Métivet, 1999), with comprehensive summaries in publications (Ladevèze & Pelle, 2004; Ladevèze & Chamoin, 2016). The following paragraphs describe the CRE concept in detail. First, the notations of the direct mechanical problem are introduced.

□ Direct mechanical problem

Let us consider a body in initial configuration $\Omega \subset \mathbb{R}^d (d = 1, 2, 3)$ with boundary $\partial\Omega$ and isothermal environment. Dirichlet boundary conditions are imposed on $\partial\Omega_1 \subset \partial\Omega$ by means of a displacement field \mathbf{u}_d . Neumann boundary conditions are prescribed on $\partial\Omega_2 \subset \partial\Omega$ by means of a force field \mathbf{f}_s^d . Here $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$. A body force field \mathbf{f}_v^d may also be prescribed in Ω .

The solution to the direct mechanical problem is the couple $(\mathbf{u}, \boldsymbol{\sigma})$ of displacement and Cauchy stress fields that satisfies the three following groups of equations:

- **kinematic admissibility** defines the space \mathcal{U}_{ad} of displacement fields satisfying the Dirichlet boundary conditions:

$$\mathbf{u}|_{\partial\Omega_1} = \mathbf{u}_d \quad (1.1)$$

- **static admissibility** defines the space \mathcal{S}_{ad} of stress fields satisfying the equilibrium:

$$\int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon}(\mathbf{v}) \, d\Omega = \int_{\Omega} \mathbf{f}_v^d \cdot \mathbf{v} \, d\Omega + \int_{\partial\Omega_2} \mathbf{f}_d^s \cdot \mathbf{v} \, dS \quad \forall \mathbf{v} \in \mathcal{U}_0 \quad (1.2)$$

with the linearized strain tensor $\boldsymbol{\epsilon}(\mathbf{v}) = \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$ and \mathcal{U}_0 the space of kinematic admissibility with homogeneous Dirichlet conditions.

- **constitutive relation** defines the space Γ of strain-stress couples that satisfy constitutive relation:

$$\boldsymbol{\sigma} = \mathbf{K} \boldsymbol{\epsilon}(\mathbf{u}) \quad (1.3)$$

The philosophy of the CRE is to divide the equations of the direct mechanical problem (1.1, 1.2, 1.3) according to the reliability of information. The constitutive relation is considered an unreliable equation, while the admissibility equations are considered reliable. For a given admissible pair $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})$, *i.e.* $\hat{\mathbf{u}} \in \mathcal{U}_{ad}$ and $\hat{\boldsymbol{\sigma}} \in \mathcal{S}_{ad}$, one can measure how much it does not satisfy the constitutive relation, thus computing the so-called constitutive relation error:

$$\mathcal{E}_{CRE}^2(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}}) = \int_{\Omega} \|\hat{\boldsymbol{\sigma}} - \mathbf{K} \boldsymbol{\epsilon}(\hat{\mathbf{u}})\|_{\mathbf{K}^{-1}}^2 \, d\Omega = \int_{\Omega} (\hat{\boldsymbol{\sigma}} - \mathbf{K} \boldsymbol{\epsilon}(\hat{\mathbf{u}})) : \mathbf{K}^{-1} : (\hat{\boldsymbol{\sigma}} - \mathbf{K} \boldsymbol{\epsilon}(\hat{\mathbf{u}})) \, d\Omega \quad (1.4)$$

While the displacement field obtained in an FE calculation naturally satisfies kinematic admissibility, this stress field is not exactly statically admissible. Therefore, to evaluate a discretization error bound with the CRE, a fully equilibrated stress field must be computed. This may be done using the hybrid-flux method (or Element Equilibration Technique - EET) (Ladevèze & Leguillon, 1983; Coorevits et al., 1992; Ladevèze & Maunder, 1996; Florentin

et al., 2002; Pled et al., 2011), flux-free methods (Pares et al., 2006; Cottureau et al., 2009; Pares et al., 2009), or Raviart-Thomas-Nédélec elements (Ern et al., 2007; Ern & Vohralik, 2010).

A visual interpretation is given in Figure 1.1. In the space of strain-stress pairs, we can see that for an admissible pair $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})$, the CRE is a distance to the space of stress-strain pairs satisfying the constitutive relation.

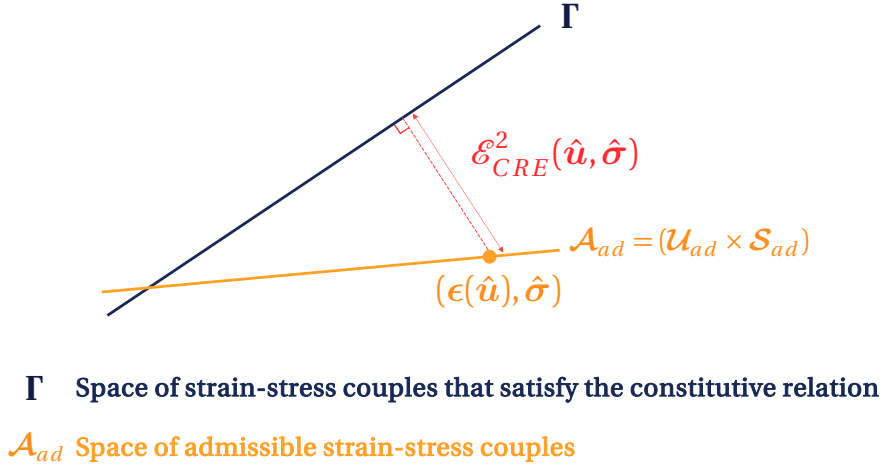


Figure 1.1 • Interpretation of the CRE in the space of strain-stress couples

1.2 The mCRE for inverse problem

In the 1990s, the CRE concept was adapted to solve inverse problems. In the context of inverse problems, in addition to the direct problems equations (1.1, 1.2, 1.3), displacement (or strain) observations are available. The inverse problem consists in finding the constitutive model parameters \mathbf{p} (e.g. Young's modulus and Poisson's ratio) that best fit the observations.

In the first approach, observations were enforced in the definition of the admissible space and optimal parameters were found by minimizing the CRE cost function (Kohn, 1988; Ladevèze & Reynier, 1989), but this approach was not well suited in the case of important measurement noise. In (Ladevèze et al., 1994), a more flexible approach (the one used in this manuscript) was proposed to enforce only reliable information on the admissible space \mathcal{A}_d , thus consistently extending the general framework. This more flexible approach, now known as the modified Constitutive Relation Error (mCRE), is described below.

$$\mathbf{p}_{opt} = \operatorname{argmin}_{\mathbf{p}} \left[\min_{(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}}) \in \mathcal{A}_d} \mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}}; \mathbf{p}) \right] \quad (1.5)$$

with

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}}; \mathbf{p}) = \int_{\Omega} \|\hat{\boldsymbol{\sigma}} - \mathbf{K}(\mathbf{p})\boldsymbol{\epsilon}(\hat{\mathbf{u}})\|_{\mathbf{K}^{-1}}^2 d\Omega + \alpha \|\Pi \mathbf{u} - \mathbf{u}_{obs}\|^2 \quad (1.6)$$

where $\mathcal{A}_d = (\mathbf{U}_{ad} \times \mathbf{S}_{ad})$, α is a scaling factor, and Π is an operator that extracts observations from the displacement field \mathbf{u} so that the model can be compared to the observations.

Equation (1.6) stands for the case where the boundary conditions are assumed to be known. This formulation can be extended for unknown boundary conditions through the addition of a term accounting for the mismatch between measured and predicted boundary conditions.

1.3 The mCRE minimization for linear elasticity

The minimization of (1.5) depends on the type of model considered but the general principle is always a 2-step iterative minimization (as summarized in Algorithm 1):

- a first step with fixed parameters $\mathbf{p}^{(n)}$, where a new admissible field pair $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})^{(n+1)}$ is found to minimize the mCRE. As we will see, this is a minimization under the admissibility constraints. The kinematic admissibility is enforced in the search space, while the static admissibility is enforced by a Lagrangian.
- a second step is performed with a fixed admissible field couple $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})^{(n+1)}$, in which the parameters are updated with a gradient descent step. The gradient of the mCRE functional with respect to the constitutive model parameters is computed using the adjoint state method.

For the sake of clarity, the minimization of (3.5) is presented here in the case of linear elasticity, and the extension of the mCRE formulation is detailed in Section 1.5.

Algorithm 1 General philosophy of mCRE minimization

- 1: Initialize parameters $\mathbf{p}^{(0)}$
 - 2: **while** not converged **do**
 - 3: **Step 1** with fixed parameters $\mathbf{p}^{(n)}$:
 - 4: Find new admissible field couple $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})^{(n+1)}$ to minimize mCRE.
 - 5: **Step 2** with fixed admissible field couple $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})^{(n+1)}$:
 - 6: Update parameters with a gradient descent step.
 - 7: **end while**
-

1.3.1 Step 1: computation of admissible fields minimizing the mCRE (for fixed parameters)

In this step parameters \mathbf{p} are fixed and the goal is to find :

$$(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}}) = \operatorname{argmin}_{(\mathbf{u}, \boldsymbol{\sigma}) \in \mathcal{A}_d} \left[\mathcal{E}_{mCRE}^2(\mathbf{u}, \boldsymbol{\sigma}; \mathbf{p}) \right] \quad (1.7)$$

First, a displacement field $\hat{\mathbf{v}}$ is introduced (from dualization) such that $\hat{\boldsymbol{\sigma}} = \mathbf{K}(\mathbf{p})\boldsymbol{\epsilon}(\hat{\mathbf{v}})$. By replacing $\hat{\boldsymbol{\sigma}}$ in (1.6), the mCRE function is written:

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \mathbf{p}) = \int_{\Omega} \|\mathbf{K}(\mathbf{p})(\boldsymbol{\epsilon}(\hat{\mathbf{v}}) - \boldsymbol{\epsilon}(\hat{\mathbf{u}}))\|_{\mathbf{K}^{-1}}^2 d\Omega + \alpha \|\Pi \mathbf{u} - \mathbf{u}_{obs}\|^2 \quad (1.8)$$

In a discretized version (following the classical FE discretization), the mCRE reads:

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{U}}, \hat{\mathbf{V}}, \mathbf{p}) = \frac{1}{2}(\hat{\mathbf{U}} - \hat{\mathbf{V}})^T \mathbb{K}(\mathbf{p})(\hat{\mathbf{U}} - \hat{\mathbf{V}}) + \alpha(\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs})^T (\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs}) \quad (1.9)$$

where the capital letters denote the discretized associated field, and \mathbb{K} is the global stiffness matrix.

In step 1, admissible fields $(\hat{\mathbf{u}}, \hat{\boldsymbol{\sigma}})$ are searched to minimize the mCRE function under the admissibility constraints. The static admissibility constraint is:

$$\int_{\Omega} \hat{\boldsymbol{\sigma}} : \boldsymbol{\epsilon}(\boldsymbol{\lambda}) d\Omega = \int_{\Omega} \mathbf{f}_d^v \cdot \boldsymbol{\lambda} d\Omega + \int_{\partial\Omega_2} \mathbf{f}_d^s \cdot \boldsymbol{\lambda} dS \quad \forall \boldsymbol{\lambda} \in \mathcal{U}_0 \quad (1.10)$$

where $\boldsymbol{\lambda}$ is a field of Lagrange multipliers.

Thus, the constrained minimization problem of Step 1 is formulated with the following discretized Lagrangian:

$$\mathcal{L}^h(\hat{\mathbf{U}}, \hat{\mathbf{V}}, \boldsymbol{\Lambda}) = \frac{1}{2}(\hat{\mathbf{U}} - \hat{\mathbf{V}})^T \mathbb{K}(\mathbf{p})(\hat{\mathbf{U}} - \hat{\mathbf{V}}) + \alpha(\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs})^T (\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs}) - \boldsymbol{\Lambda}^T (\mathbb{K}(\mathbf{p})\hat{\mathbf{V}} - \mathbf{F}) \quad (1.11)$$

where \mathbf{F} is the global load vector. To impose the kinematic admissibility, with finite element discretization, only the free degrees of freedom (dof) are searched for, and the prescribed dofs are imposed in the search space.

Remark

Section 1.1 introduced the concept of CRE in the context of FE verification, where the CRE serves as a bound on the discretization error. In this context, the CRE is computed with a post-processed $\hat{\boldsymbol{\sigma}}$ from the EET method (or other methods). On the contrary, in the context of inverse problems, the discretization error is usually negligible compared to the error caused by wrong model parameters. The admissible field $\hat{\boldsymbol{\sigma}}$ used in the calculation of the mCRE functional is directly the field obtained from the FE discretization and is not post-processed (cf (Nguyen, 2021) to assess the discretization error).

The search for the stationarity of the Lagrangian leads to the following system:

$$\begin{cases} \frac{\partial \mathcal{L}^h}{\partial \hat{\mathbf{U}}} = \mathbf{0} \\ \frac{\partial \mathcal{L}^h}{\partial \hat{\mathbf{V}}} = \mathbf{0} \\ \frac{\partial \mathcal{L}^h}{\partial \boldsymbol{\Lambda}} = \mathbf{0} \end{cases} \Leftrightarrow \begin{cases} \mathbb{K}(\mathbf{p})(\hat{\mathbf{U}} - \hat{\mathbf{V}}) + \alpha \Pi^T (\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs}) = \mathbf{0} \\ \mathbb{K}(\mathbf{p})(\hat{\mathbf{V}} - \hat{\mathbf{U}}) - \mathbb{K}(\mathbf{p})\boldsymbol{\Lambda} = \mathbf{0} \\ \mathbb{K}(\mathbf{p})\hat{\mathbf{V}} - \mathbf{F} = \mathbf{0} \end{cases} \quad (1.12)$$

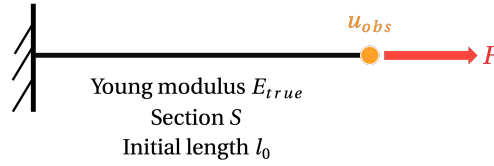
From $\frac{\partial \varphi^h}{\partial \hat{\mathbf{V}}} = \mathbf{0}$ it comes $\mathbf{\Lambda} = \hat{\mathbf{V}} - \hat{\mathbf{U}}$, and (1.12) eventually yields:

$$\begin{cases} \mathbb{K}(\mathbf{p})\hat{\mathbf{V}} = \mathbf{F} \\ (\mathbb{K}(\mathbf{p}) + \alpha\Pi^T\Pi)\hat{\mathbf{U}} = \mathbf{F} + \alpha\Pi^T\mathbf{U}_{obs} \end{cases} \quad (1.13)$$

(1.13) is fundamental in the mCRE method. It shows that the field $\hat{\mathbf{V}}$ is obtained from the FEM solution of the direct problem (Block 1.1) for a given set of material parameters, while $\hat{\mathbf{U}}$ is a compromise between the model and the observations (weighted by the parameter α). In practice, one way to find a meaningful value of α is to choose it so that $\hat{\mathbf{U}}$ fits the observations up to the measurement noise. This technique is called the Morozov criterion (Morozov, 1968), as further developed in Chapter 2.

The following example illustrates the consequence of this on the interpretation of the mCRE terms.

Example: Let us consider a one-dimensional beam, with section S , initial length l_0 , and Young modulus E_{true} . The constitutive relation is the Hooke law $\sigma = E_{true}\epsilon$. The beam is clamped at $x = 0$ and is loaded in tension with a known force F . For the sake of simplicity, this problem is limited to one degree of freedom, where the displacement u_{obs} at the end of the beam is observed.



Let us consider the case where $E_{guess} \neq E_{true}$ in a situation without measurement noise. System (1.13) reads:

$$\begin{cases} K_{guess}\hat{v} = F \\ (K_{guess} + \alpha)\hat{u} = F + \alpha u_{obs} \end{cases} \Leftrightarrow \begin{cases} \hat{v} = \frac{F}{K_{guess}} \\ \hat{u} = \frac{F + \alpha u_{obs}}{K_{guess} + \alpha} \end{cases} \quad (1.14)$$

with $K_{guess} = \frac{E_{guess}S}{l_0}$. To understand the behavior of the mCRE functional, the following details the case $\alpha \rightarrow \infty$ and $\alpha = 0$

When $\alpha \rightarrow \infty$:

In this case, strong importance is given to the discrepancy of the data in the mCRE functional (1.6). Therefore, the displacement \hat{u} obtained at the end of Step 1 tends to the observed displacement u_{obs} , which can be observed in (1.14). In this case, when the mCRE is computed at the end of Step 1, the term $\alpha\|\Pi\mathbf{u} - \mathbf{u}_{obs}\|^2$ is zero and only the CRE term appears in the mCRE functional:

$$\mathcal{E}_{mCRE}^2(\hat{u}, \hat{v}; E_{guess}) = \frac{1}{2} K_{guess} (\hat{v} - \hat{u})^2 \quad (1.15)$$

$$= \frac{1}{2} K_{guess} \left(\frac{F}{K_{guess}} - u_{obs} \right)^2 \quad (1.16)$$

with $u_{obs} = \frac{Fl_0}{E_{true}S}$. The further E_{guess} from E_{true} , the larger the CRE term. This observation justifies the name of "modeling error" given to the CRE term.

When $\alpha = 0$:

In this case, the modeling error term in the mCRE function (1.6) is strongly emphasized, and \hat{u} is forced to satisfy the assumed model: $\hat{u} = \frac{F}{K_{guess}} = \hat{v}$. Therefore, when the mCRE is computed with the value of \hat{u} and \hat{v} obtained at the end of Step 1, the modeling error is 0. This observation may seem counterintuitive since one can say that there should be a modeling error as soon as E_{guess} is not E_{true} . Actually, the chosen value α does not make this modeling error appear, because the solution \hat{u} is not forced to compromise with the observation.

The modeling error should here be understood as the error in the model structure (in this example, the structure of the model is correct, and the model parameter is incorrect).

1.3.2 Step 2: updating model parameters with a gradient descent step

This step consists of the updating of the parameters \mathbf{p} with a gradient descent step:

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} - l_r \frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{U}}^{(n+1)}, \hat{\mathbf{V}}^{(n+1)}; \mathbf{p}^{(n)})}{d\mathbf{p}} \quad (1.17)$$


where $(\hat{\mathbf{U}}^{(n+1)}, \hat{\mathbf{V}}^{(n+1)})$ are the fields obtained at the end of Step 1 of the $n + 1$ iteration of the mCRE minimization. For simplicity, $(\hat{\mathbf{U}}^{(n+1)}, \hat{\mathbf{V}}^{(n+1)})$ will be denoted as $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$. The gradient of mCRE with respect to the parameters is computed using the adjoint-state method:

$$\frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{U}}, \hat{\mathbf{V}}; \mathbf{p})}{d\mathbf{p}} = \frac{d\mathcal{L}(\hat{\mathbf{U}}, \hat{\mathbf{V}}, \lambda; \mathbf{p})}{d\mathbf{p}} \quad (1.18)$$

$$= \frac{d\hat{\mathbf{U}}}{d\mathbf{p}} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{U}}} + \frac{d\hat{\mathbf{V}}}{d\mathbf{p}} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{V}}} + \frac{d\lambda}{d\mathbf{p}} \frac{\partial \mathcal{L}}{\partial \lambda} + \frac{\partial \mathcal{L}}{\partial \mathbf{p}} \quad (1.19)$$

$$= \frac{\partial \mathcal{L}}{\partial \mathbf{p}} \quad (1.20)$$


$$\frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{U}}, \hat{\mathbf{V}}; \mathbf{p})}{d\mathbf{p}} = \frac{1}{2} (\hat{\mathbf{U}} - \hat{\mathbf{V}})^T \frac{\partial \mathbb{K}(\mathbf{p})}{\partial \mathbf{p}} (\hat{\mathbf{U}} + \hat{\mathbf{V}}) \quad (1.21)$$

 **Example:** In the previous example, this is the gradient of the mCRE with respect to the model parameters, here E_{guess} :

$$\frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{U}}, \hat{\mathbf{V}}; \mathbf{p})}{d\mathbf{p}} = \frac{S}{2l_0}(\hat{u} - \hat{v})(\hat{u} + \hat{v}) \tag{1.22}$$

$$\tag{1.23}$$

which is 0 if $E_{guess} = E_{true}$ (without noise). In the case where $E_{guess} \neq E_{true}$, if $\alpha = 0$ there is no difference between \hat{u} and \hat{v} , so the Young's modulus is not updated: it is important to choose a value that creates a modeling error term. It is also worth noting that α and the learning rate l_r both influence the optimization process. The larger the α , the larger the difference between \hat{u} and \hat{v} , the larger the update of the model parameters.

 **Online tool:** The previous example is implemented in Python and is available in the GitHub repository in the file `Chap_1/linear_mCRE.ipynb`. It may be interesting to analyze the influence of the noise level, the value of α and the initial guess E_{guess} on \hat{u} and the value of the mCRE functional.

1.4 Previous works on mCRE

The mCRE framework - also referred to as the Modified Error in Constitutive Equations (MECE) - is now widely studied in diverse applications as detailed in Table 1.1

| Application | References |
|-----------------------------|--|
| Dynamics | (Ladevèze et al., 1994; Chouaki et al., 1996; Ladevèze & Chouaki, 1999; Deraemaeker et al., 2002; Allix et al., 2003; Barthe et al., 2004; Deraemaeker et al., 2004; Allix et al., 2005; Ladevèze et al., 2006; Feissel & Allix, 2007; Faverjon & Sinou, 2008; Nguyen et al., 2008; Banerjee et al., 2013; Charbonnel et al., 2013; Bonnet & Aquino, 2015; Diaz et al., 2015; Guchhait & Banerjee, 2016; Silva & Maia, 2017; Guchhait & Banerjee, 2018; Diaz et al., 2022) |
| Acoustics | (Decouvreur et al., 2004, 2007, 2008; Warner et al., 2014) |
| Corrupted measurements | (Allix et al., 2005; Feissel & Allix, 2007; Nguyen et al., 2008) |
| Full-field measurements | (Ben Azzouna et al., 2015; Huang et al., 2016; Ghosh et al., 2017; Banerjee et al., 2013; Guchhait & Banerjee, 2016, 2018; Ferrier et al., 2021; Nguyen, 2021) |
| In situ measurements | (Charbonnel et al., 2013; Bouclier et al., 2013) |
| Defect detection | (Bui & Constantinescu, 2000; Faverjon et al., 2009; Waeytens et al., 2016; Barbarella et al., 2016; Hu et al., 2017, 2019; Diaz et al., 2024) |
| Nonlinear material behavior | (Chouaki et al., 1998; Hadj-Sassi, 2007; Guchhait & Banerjee, 2015; Marchand et al., 2019; Nguyen, 2021) |

Table 1.1 • mCRE applications with corresponding references

This wide range of works has highlighted several interesting properties of the mCRE framework, such as improved convexity with respect to the parameters to be identified compared to other functionals (Bonnet & Constantinescu, 2005; Feissel & Allix, 2007; Diaz, 2023), spatial localization of errors (Bui & Constantinescu, 2000; Ben Azzouna et al., 2015; Barbarella et al., 2016), robustness to noise on the observations (Allix et al., 2005; Feissel & Allix, 2007; Nguyen et al., 2008), or the ability to deal with partially known boundary conditions (Diaz et al., 2015; Bonnet & Aquino, 2015; Aquino & Bonnet, 2019; Ferrier et al., 2021).

Remark

A stochastic interpretation of mCRE was proposed in (Deraemaeker et al., 2004). Since covariance on the modeling error is usually not known, the idea is to integrate the CRE term (used at modeling error) into Bayesian inference. The value of the CRE term is then used to quantify the confidence of the model, accounting for modeling error.

The present manuscript builds on and extends the work on parameter identification of nonlinear constitutive laws. The following section recalls the mCRE formulation in the case of nonlinear behavior within the framework of the Generalized Standard Material (Halphen & Nguyen, 1975).

1.5 The mCRE framework for nonlinear behavior

The extension of mCRE to nonlinear material behavior is based on the thermodynamically consistent framework of the Generalized Standard Material. This framework is first recalled before the mCRE extensions for nonlinear behavior are presented.

1.5.1 Thermodynamics framework

This section aims to describe the thermodynamic framework of the Generalized Standard Material (Halphen & Nguyen, 1975) used in this work. In such a formulation, the material behavior is described in terms of state equations and evolution laws involving convex potential, pseudo-potential, and internal variables. The following recalls the properties that should be satisfied to be automatically compatible with the principles of thermodynamics. A more detailed explanation can be found in (Lemaitre & Chaboche, 1990).

First principle

The conservation of energy is described by the first principle of thermodynamics. The first principle of thermodynamics locally reads:

$$\rho \dot{e} = \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} + r - \nabla \cdot \mathbf{q} \quad (1.24)$$

with the specific energy density e , the Cauchy stress tensor $\boldsymbol{\sigma}$, the linearized strain tensor $\boldsymbol{\epsilon}$, the volume heat source r , and the heat flux \mathbf{q} .

Second principle

The second principle of thermodynamics reads locally:

$$\rho \dot{s} + \nabla \cdot \left(\frac{\mathbf{q}}{\theta} \right) - \frac{r}{\theta} \geq 0 \quad (1.25)$$

with s the specific entropy density and θ the temperature.

Clausius-Duhem inequality

After introducing the Helmholtz free energy $\psi = \rho(e - \theta s)$, the Clausius-Duhem inequality is obtained by combining (1.24) and (1.25):

$$\boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} - \dot{\psi} - \frac{\mathbf{q} \cdot \nabla \theta}{\theta} \geq 0 \quad (\text{for an isothermal process}) \quad (1.26)$$

This inequality is fundamental for defining thermodynamically consistent constitutive laws. For a problem with plasticity, the total strain tensor is written with the following decomposition: $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_e + \boldsymbol{\epsilon}_p$, including the elastic strain $\boldsymbol{\epsilon}_e$ and the plastic strain $\boldsymbol{\epsilon}_p$. Internal variables X_k ($k = 1, \dots, K$), collected in the vector \mathbf{X} , are introduced to represent other phenomena (such as hardening). The state equations are derived from the Helmholtz free energy, which is used as the thermodynamic potential $\psi(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}_p, \mathbf{X})$. The Clausius-Duhem inequality can be rewritten:

$$\left(\boldsymbol{\sigma} - \frac{\partial \psi}{\partial \boldsymbol{\epsilon}_e} \right) : \dot{\boldsymbol{\epsilon}}_e + \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}_p - \sum_{k=1}^K \frac{\partial \psi}{\partial X_k} \dot{X}_k - \frac{\mathbf{q} \cdot \nabla \theta}{\theta} \geq 0 \quad (1.27)$$

with $\left(\boldsymbol{\sigma} - \frac{\partial \psi}{\partial \boldsymbol{\epsilon}_e} \right) : \dot{\boldsymbol{\epsilon}}_e + \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}_p - \sum_{k=1}^K \frac{\partial \psi}{\partial X_k} \dot{X}_k$ the intrinsic dissipation. The use of a convex function ψ automatically satisfies the Clausius-Duhem inequality. For reversible elastic transformation with homogeneous temperature, it comes $\boldsymbol{\sigma} = \frac{\partial \psi}{\partial \boldsymbol{\epsilon}_e}$ corresponding to the Hooke's law. By analogy, thermodynamic forces Y_k gathered in \mathbf{Y} are defined such that they are associated with the internal variable vector \mathbf{X} with $\mathbf{Y} = \frac{\partial \psi}{\partial \mathbf{X}}$.

A convex, non-negative and zero at origin dissipation pseudo-potential $\varphi(\dot{\boldsymbol{\epsilon}}_p, \dot{\mathbf{X}})$ may be postulated for the definition evolution laws, to guarantee the positivity of the intrinsic dissipation $\mathcal{D} = (\boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}_p - \mathbf{Y} : \dot{\mathbf{X}})$. The evolution laws are derived from the pseudo-potential with:

$$\boldsymbol{\sigma} = \frac{\partial \varphi}{\partial \dot{\boldsymbol{\epsilon}}_p} \mathbf{Y} = - \frac{\partial \varphi}{\partial \dot{\mathbf{X}}} \quad (1.28)$$

📖 **Generalized Standard Material framework**

To summarize, a way to model a constitutive behavior in the thermodynamically consistent framework of Generalized Standard Material (Halphen & Nguyen, 1975) is to establish the functions ψ and φ respectively involved in the set of state laws and evolutions laws and satisfying the following properties:

- ψ is convex;
- φ is convex, non-negative and zero at origin.

The dual potentials are defined by the Legendre-Fenchel transform:

$$\psi^*(\mathbf{s}) = \sup_{\mathbf{e}_e} [\mathbf{s} \cdot \mathbf{e}_e - \psi(\mathbf{e}_e)] \quad \text{and} \quad \varphi^*(\mathbf{s}) = \sup_{\dot{\mathbf{e}}_p} [\mathbf{s} \cdot \dot{\mathbf{e}}_p - \varphi(\dot{\mathbf{e}}_p)] \quad (1.29)$$

with $\mathbf{e}_e = [\boldsymbol{\epsilon}_e, \mathbf{X}]$, $\mathbf{e}_p = [\boldsymbol{\epsilon}_p, -\mathbf{X}]$ the global flux variables, and $\mathbf{s} = [\boldsymbol{\sigma}, \mathbf{Y}]$ the global thermodynamic forces. The potential ψ , φ , ψ^* and φ^* are involved in the definition of the Constitutive Relation Error (CRE) recalled in Section 1.5.2.

📖 **Remark**

For a rate-independent behavior, φ^* is not differentiable and the yield criterion function $f(\boldsymbol{\sigma})$ is introduced. In this case:

$$\varphi^*(\boldsymbol{\sigma}) = I(f) = \begin{cases} 0 & \text{if } f < 0 \\ +\infty & \text{if } f = 0 \end{cases} \quad (1.30)$$

1.5.2 mCRE formulation for nonlinear behavior

In the case of a nonlinear material behavior, the mCRE functional still consists of a CRE term and a discrepancy to the observations term. For an admissible solution $\hat{\mathbf{s}} = (\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\epsilon}}_p, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}})$, the mCRE is:

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{s}}; \mathbf{p}) = \mathcal{E}_{CRE}^2(\hat{\mathbf{s}}; \mathbf{p}) + \frac{\alpha}{2} \int_0^T \|\Pi \mathbf{u} - \mathbf{u}_{obs}\|^2 dt \quad (1.31)$$

in which the CRE term is composed of a term η_ψ involving the free-energy and a term η_φ taking into account the dissipation (Ladevèze & Moës, 1998; Ladevèze, 2001; Ladevèze & Pelle, 2004):

$$\mathcal{E}_{CRE}^2(\hat{\mathbf{s}}) = \int_0^T \int_\Omega \eta_\psi(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) d\Omega dt + \int_0^T \int_0^t \int_\Omega \eta_\varphi(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) d\Omega ds dt \quad (1.32)$$

with

$$\eta_\psi(\hat{\epsilon}_e, \hat{\sigma}, \hat{X}, \hat{Y}) = \psi(\hat{\epsilon}_e, \hat{X}) + \psi^*(\hat{\sigma}, \hat{Y}) - \hat{\sigma} : \hat{\epsilon}_e - \hat{X} : \hat{Y} \quad (1.33)$$

$$\eta_\varphi(\hat{\epsilon}_p, \hat{\sigma}, \hat{X}, \hat{Y}) = \varphi(\hat{\epsilon}_p, -\hat{X}) + \varphi^*(\hat{\sigma}, \hat{Y}) - \hat{\sigma} : \hat{\epsilon}_p + \hat{X} : \hat{Y} \quad (1.34)$$

Remark

As in the linear case, the CRE functional (1.32) has been used as an FE element verification in a wide range of applications (Ladevèze & Moës, 1997; Chamoin & Ladevèze, 2008; Ladevèze & Chamoin, 2010; Chamoin et al., 2012; Waeytens et al., 2012; Ladevèze et al., 2012; Chamoin et al., 2017).

Figure 1.2 shows an interpretation of (1.33) defined by the term η_ψ for a constitutive law without dissipation and without internal variable. For a given point $(\hat{\epsilon}, \hat{\sigma})$, the area in blue is $\psi^*(\hat{\sigma})$, the area in grey is $\psi(\hat{\epsilon})$, the area bounded by the red rectangle is $\hat{\sigma} : \hat{\epsilon}$. So the hatched area is $\eta_\psi(\hat{\epsilon}, \hat{\sigma}) = \psi(\hat{\epsilon}) + \psi^*(\hat{\sigma}) - \hat{\sigma} : \hat{\epsilon}$.

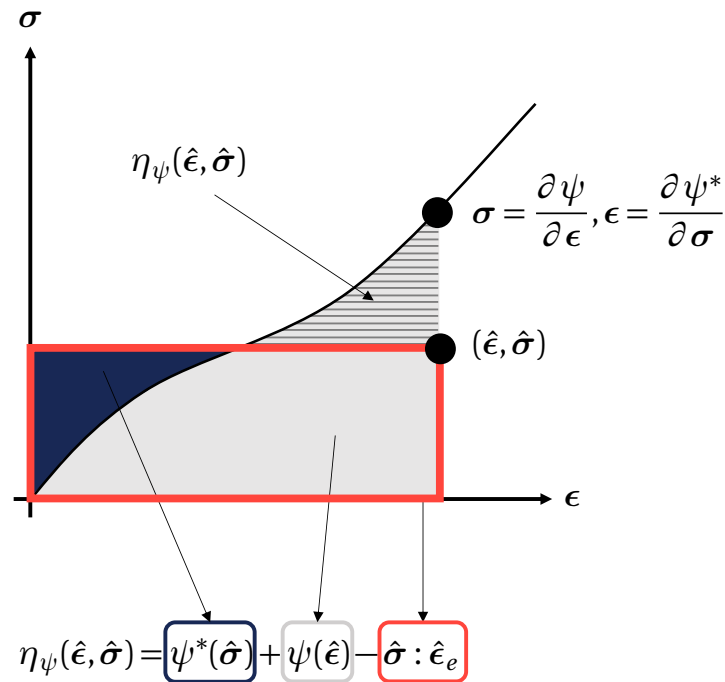


Figure 1.2 • Interpretation of the term η_ψ defined in (1.33).

Remark

Without dissipation and with quadratic potential ψ describing the Hooke law ($\psi(\epsilon) =$

$\frac{1}{2} \boldsymbol{\epsilon} : \mathbf{K} \boldsymbol{\epsilon}$ and $\psi^*(\boldsymbol{\sigma}) = \frac{1}{2} \boldsymbol{\sigma} : \mathbf{K}^{-1} \boldsymbol{\sigma}$, the linear elasticity constitutive relation is $\boldsymbol{\sigma} = \mathbf{K} : \boldsymbol{\epsilon}$ and the CRE is the same as the one written in (1.4).

The minimization of the mCRE for nonlinear constitutive laws remains close to the idea of Algorithm 1 presented in the linear case. The main modification is the way the admissible fields are computed, since evolution laws have to be integrated. In (Marchand et al., 2019; Nguyen, 2021), Step 1 is realized with a strategy inspired by the LATIN method (Ladevèze, 1999), which is non-incremental (*i.e.* global in time) and well suited to the mathematical structure of the mCRE. The choice is to split the mCRE into two positive parts $\mathcal{E}_\psi^2(\boldsymbol{\epsilon}_e, \boldsymbol{\sigma}, \mathbf{X}, \mathbf{Y})$ and $\mathcal{E}_\varphi^2(\dot{\boldsymbol{\epsilon}}_p, \boldsymbol{\sigma}, \dot{\mathbf{X}}, \mathbf{Y})$ defined by:

$$\mathcal{E}_\psi^2(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) = \int_0^T \int_\Omega \eta_\psi(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) d\Omega dt + \frac{\alpha}{2} \int_0^T \|\Pi \hat{\mathbf{u}} - \mathbf{u}_{obs}\|^2 dt \quad (1.35)$$

$$\mathcal{E}_\varphi^2(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \dot{\hat{\mathbf{X}}}, \hat{\mathbf{Y}}) = \int_0^T \int_0^t \int_\Omega \eta_\varphi(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \dot{\hat{\mathbf{X}}}, \hat{\mathbf{Y}}) d\Omega ds dt \quad (1.36)$$

This separation into two positive parts allows the minimization of each term alternatively. Equation (3.9) is a compromise between the residual on the state equations and the discrepancy with measurements. Its minimization is a linear (because elasticity is assumed to be linear here) and global in space problem. The minimization of (3.10) is local in space and corresponds to the integration of the evolution laws, here performed with an Euler scheme. This strategy is maintained in the present manuscript and further details are given in Chapter 3. In all previous works on the mCRE framework, the form of the constitutive relation was always considered reliable (only the parameters were considered unreliable and thus updated). However, the form of the constitutive relation is not always reliable in practice, which is referred to as model bias. In the remainder of the manuscript, the form of the constitutive relation is relaxed by using neural networks to describe the thermodynamic potentials ψ and φ .

2 Coupling Deep Learning and constitutive modeling

2.1 Introduction to Deep Learning

Before delving into the specificities of this research, this section positions Deep Learning in the context of artificial intelligence and provides a brief general description of the basic principles of Deep Learning. As presented in Figure 1.3, Deep Learning is a subset of Machine Learning which is also a subset of Artificial Intelligence.

Artificial Intelligence, often abbreviated as AI, is the emulation of human intelligence processes by computer systems. It encompasses a wide range of techniques and methodologies designed to enable computers to perform tasks that typically require human intelligence. These tasks include decision-making, knowledge representation, and rule-based systems.

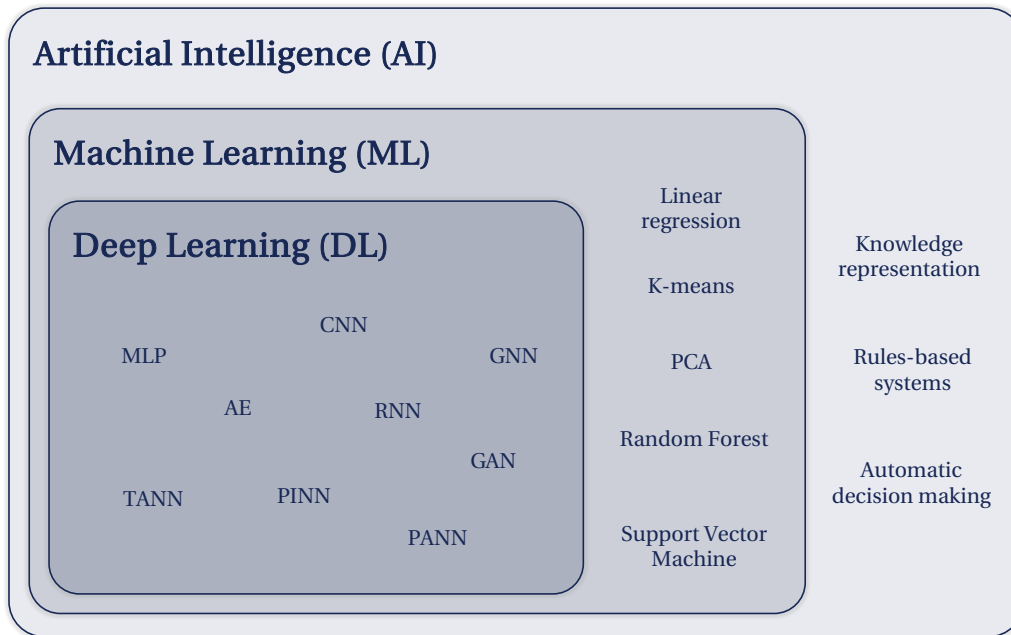


Figure 1.3 • Classification of Artificial Intelligence methods.

AI systems can be designed to operate autonomously or with human supervision.

Machine learning (ML) is a subset of artificial intelligence that focuses on developing algorithms and statistical models that enable computers to learn from data and make predictions or decisions based on that data, without being explicitly programmed to perform specific tasks. In other words, Machine Learning algorithms learn patterns and relationships from large data sets, allowing them to generalize and make predictions or decisions about new, unseen data. ML algorithms can be categorized into supervised learning, unsupervised learning, and reinforcement learning, depending on the type of training data and learning goals. Common applications of Machine Learning include classification, regression, clustering, dimensionality reduction, and recommendation systems, involving methods such as linear regression, K-means, Principal Component Analysis (PCA), Random Forest or Support Vector Machine (SVM) (Bishop, 2007).

Deep Learning is a subfield of Machine Learning that focuses on training artificial neural networks with multiple layers of interconnected nodes, also known as artificial neural networks (ANNs). Deep learning models are able to automatically learn representations of data through these multiple layers, enabling them to capture intricate patterns and features in complex datasets. Unlike traditional Machine Learning approaches that require manual feature engineering, Deep Learning algorithms can learn representations directly from raw data, making them highly versatile and suitable for a wide range of tasks. Deep Learning has achieved remarkable success in various domains, including image recognition (He et al., 2016; Krizhevsky et al., 2017), speech recognition (Amodei et al., 2016; Hinton et al., 2012), language translation (Vaswani et al., 2017; Wu et al., 2016), and autonomous driving (Bojarski et al., 2016; Chen et al., 2015), driving significant advances in the field of artificial intelligence. One reason behind the success of Deep Learning relies on the use of important datasets, coming from the internet and social networks for example (Deng et al., 2009).

As shown in Figure 1.3, Deep Learning is composed of a wide range of neural network architectures, which are briefly introduced below, designed for specific types of input data.

2.1.1 The multilayer perceptron

A multilayer perceptron (MLP), also called a feedforward neural network (FFNN), is the most common Deep Learning model (Goodfellow et al., 2016). An MLP is a mathematical model f_{NN} whose goal is to approximate a function f . The function f_{NN} is parameterized by θ , the parameters of the neural network (consisting of weights and biases). The parameters θ are adjusted to find the best approximation of the function in a process called neural network training, which is described in detail in Section 2.1.3. The output of an MLP is obtained by a series of compositions of elementary operations involving a perceptron.

A perceptron to mimic biological neurons

The building block of an MLP is the perceptron, introduced in (McCulloch & Pitts, 1943) to model the behavior of a biological neuron, and further extended and implemented by (Rosenblatt, 1958). The output of a perceptron h is defined by the following operation:

$$h = \mathcal{A}(\mathbf{W}\mathbf{x}^T - b) \quad (1.37)$$

where \mathcal{A} is an activation function (originally the Heaviside function), \mathbf{x} is the real-valued input vector, \mathbf{W} is a vector of real-valued weights, and b is a bias. The weights allow the influence of each component of the input x to be amplified, while the bias shifts the decision boundary away from its origin and remains independent of any input value.

Figure 1.4 shows an illustration of the perceptron implemented by Rosenblatt in 1958, as well as the function h of (1.37).

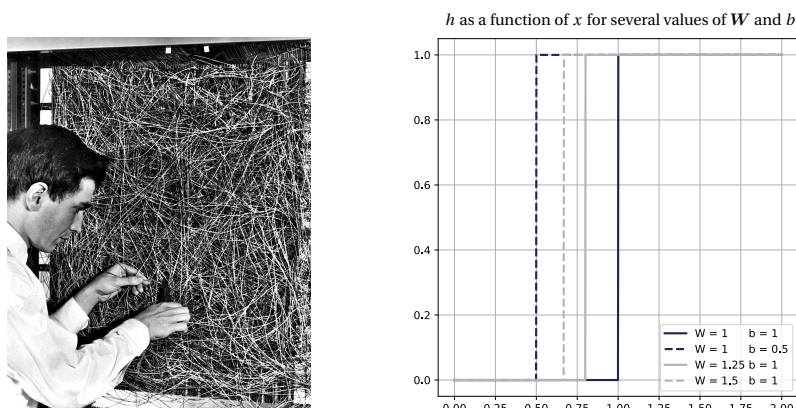


Figure 1.4 • Rosenblatt's perceptron. Left: Physical implementation of the perceptron in 1960. Right: output of a perceptron for different values of weight and bias for a scalar input.

From the perceptron to the MLP

An MLP is nothing more than an organization in a network of multiple perceptrons, as shown in Figure 1.5. Mathematically, the output of an n -layer MLP is given by

$$f_{NN}(\mathbf{x}) = \mathbf{W}_n h_{n-1}(\mathbf{W}_{n-1} h_{n-2}(\dots \mathbf{W}_2 h_1(\mathbf{W}_1 \mathbf{x}^T - \mathbf{b}_1)^T - \mathbf{b}_2)^T - \mathbf{b}_{n-1})^T - \mathbf{b}_n \quad (1.38)$$

Note that in (1.37) h is a scalar function, while in (1.38) h is applied component-wise to a vectorial input. In the example of Figure 1.5, \mathbf{x} is a 6-component line vector, \mathbf{W}_1 is a 4 by 6 matrix, \mathbf{W}_2 is a 2 by 4 matrix, \mathbf{W}_3 is a 3 by 2 matrix, \mathbf{W}_4 is a 1 by 3 matrix, \mathbf{b}_1 a 4 by 4 line vector, \mathbf{b}_2 a 2 by 2 line vector, \mathbf{b}_3 a 3 by 3 line vector, and \mathbf{b}_4 a scalar.

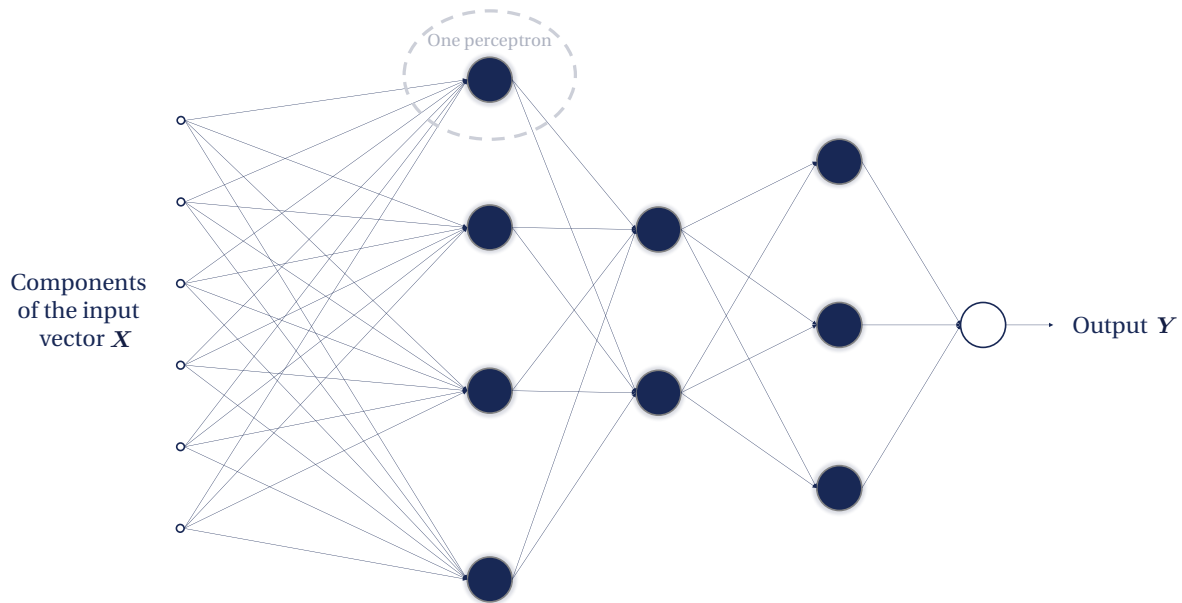


Figure 1.5 • Multilayer perceptron (MLP) in a configuration with a 6-component vector in input and respectively 4-2-3-1 perceptrons in each layer.

Example: Representing an XOR (exclusive or) with an MLP

This example illustrates how an MLP can represent an XOR function. In the XOR function, the output is 1 if the inputs are different, and 0 if the inputs are the same, the truth table of an XOR function is as follows:

| Input A | Input B | Output |
|---------|---------|--------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

The input layer consists of two neurons corresponding to the value of the input data. The hidden layer has two neurons and uses the Heaviside activation function. The weights and biases of this layer are \mathbf{W}_1 and \mathbf{b}_1 . The output layer has one neuron and also uses the Heaviside activation function. The weights and biases of this layer are \mathbf{W}_2 and \mathbf{b}_2 . One way to represent

an XOR function is to have the following parameters:

$$\mathbf{W}_1 = \begin{bmatrix} 2 & -2 \\ 2 & -2 \end{bmatrix} \quad \mathbf{b}_1 = \begin{bmatrix} -1 \\ 3 \end{bmatrix} \quad \mathbf{W}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \mathbf{b}_2 = -3$$

Universal Approximation Theorem

The Universal Approximation Theorem for neural networks, a cornerstone result in the field of Deep Learning, states that feedforward neural networks with a single hidden layer containing a finite number of neurons can approximate any continuous function on a compact input space with arbitrary accuracy, given a sufficiently large number of neurons. This theorem, first proved by Cybenko in 1989 (Cybenko, 1989) and later generalized by Hornik in 1991 (Hornik, 1991), highlights the remarkable representational power of neural networks as universal function approximators. By using a nonlinear activation function in the hidden layer, such as the sigmoid, neural networks are able to capture complex input-output mappings and learn highly nonlinear relationships within the data. The Universal Approximation Theorem provides the theoretical justification for the widespread use of neural networks in various domains. Despite its theoretical importance, this theorem is only an existence theorem and does not say anything about the ability to perform successful training, mainly because of the non-convex nature of neural network optimization, as discussed in Section 2.1.3.

2.1.2 Other network architectures

In addition to multilayer perceptrons (MLPs), the landscape of neural network architectures encompasses a wide variety of powerful models. These include the following architecture families:

- **Recurrent Neural Network (RNN):** A type of neural network designed to process sequential data, such as time series or natural language. RNNs have recurrent connections that allow information to persist over time, making them useful for tasks such as speech recognition, language modeling, and sentiment analysis (Hochreiter & Schmidhuber, 1997). The use of internal memory in RNN can be compared with the use, in mechanics, of description with internal variables.
- **Convolutional Neural Network (CNN):** A specialized type of neural network commonly used to analyze visual images. CNNs consist of multiple layers of convolutional filters that can automatically learn hierarchical representations of features. They are widely used in tasks such as image classification, object detection, and image segmentation (LeCun et al., 1998).
- **Autoencoder (AE):** A type of neural network used for unsupervised learning of efficient encodings or representations of input data. Autoencoders consist of an encoder network that compresses the input into a latent representation, and a decoder network that reconstructs the input from the latent representation. They are useful for tasks such as data denoising, dimensionality reduction, and anomaly detection (Hinton &

Salakhutdinov, 2006). In the context of mechanics, this technique shares similarities with the homogenization and localization approach.

- Graph Neural Network (GNN): A type of neural network designed to operate on graph-structured data, where nodes represent entities and edges represent relationships between them. GNNs can learn representations of nodes and graphs, enabling tasks such as node classification, graph classification, and link prediction. They are applicable in domains such as social network analysis, recommendation systems, and drug discovery (Scarselli et al., 2008). In mechanics, the use of Finite Element meshes can be compared with the GNN way to structure information.

2.1.3 Gradient-based optimization for neural network training

This section aims to give a quick overview of the optimization used in Deep Learning. More detailed explanations can be found in (Goodfellow et al., 2016). To find the optimal parameters θ of neural network architectures, gradient-based optimization methods are commonly used. These methods aim to minimize a predefined non-convex loss function $\mathcal{L}(\theta)$ that quantifies the difference between the network's predictions and the true targets in the training data. Gradient-based optimization algorithms, such as variants of gradient descent, utilize the gradient of the loss function with respect to θ computed with automatic differentiation. By iteratively updating the parameters θ in the direction of the negative gradient, these methods attempt to navigate the complex landscape of the loss function and converge to satisfactory solutions. Stochastic gradient descent (SGD) (Robbins & Monro, 1951), a variant of gradient descent, is commonly used in neural network training and is described in detail in Algorithm 2. The term "stochastic" refers to the fact that instead of computing the gradient of the loss function over the entire training data set, SGD computes it over randomly selected mini-batches of data. The size of each mini-batch, called the batch size, is a hyperparameter that can be adjusted based on computational constraints and optimization performance. An epoch, in this context, represents one complete pass through the entire training dataset. Multiple epochs are performed to improve the model gradually, but care must be taken to avoid overfitting, which occurs when the model learns the specific details and noise in the training data to an extent that it negatively impacts the performance on new data. A validation dataset, composed of different data from the training dataset, plays a crucial role in evaluating model performance during training to guard against overfitting.

The most common loss function used in neural network training is the mean squared error (MSE), which measures the average squared difference between the predicted and actual values. We also encounter alternative loss functions such as binary cross-entropy loss for binary classification tasks, categorical cross-entropy loss, or more sophisticated loss. During backpropagation, gradients of the loss function with respect to the parameters are computed using automatic differentiation, which efficiently applies the chain rule of calculus to propagate gradients through the network layers. It is important to note that gradient-based optimization methods, including stochastic gradient descent (SGD), can converge to local minima of the loss function, resulting in suboptimal solutions. Therefore,

Algorithm 2 Neural Network Training with Stochastic Gradient Descent


```

1: Input: Training dataset  $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$ , learning rate  $l_r$ , number of epochs  $n_{\text{epochs}}$ , batch
   size  $B$ 
2: Initialize neural network parameters randomly:  $\theta$ 
3: for  $t = 1$  to  $n_{\text{epochs}}$  do
4:   for  $b = 1$  to  $\frac{N}{B}$  do ▷ Iterate over mini-batches
5:     Select randomly a mini-batch  $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^B$ 
6:     Forward pass for the mini-batch:
7:       Compute neural network output:  $\hat{\mathbf{y}}^{(i)} = f_{NN}(\mathbf{x}^{(i)}, \theta)$ 
8:       Compute average loss over the mini-batch:  $\mathcal{L}^{(b)} = \frac{1}{B} \sum_{i=1}^B \text{Loss}(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$ 
9:       Backward pass (backpropagation) for the mini-batch:
10:      Compute gradients:  $\nabla_{\theta} = \nabla_{\theta} \mathcal{L}^{(b)}$ 
11:      Update parameters using stochastic gradient descent (SGD):
12:       $\theta \leftarrow \theta - l_r \nabla_{\theta} \mathcal{L}^{(b)}$ 
13:   end for
14: end for
15: Output: Trained neural network parameters:  $\theta$ 

```

multiple initializations and careful tuning of hyperparameters are often required to mitigate the risk of getting stuck in local minima.

Beyond the SGD presented here, adaptive learning rate methods such as AdaGrad (Duchi et al., 2011), RMSProp (Tieleman & Hinton, 2012), and Adam (Kingma & Ba, 2015) dynamically adjust the learning rate during training, providing faster convergence and improved performance on non-convex optimization problems with varying curvature.

 **Online tool:** Neural Networks playground:

<https://playground.tensorflow.org> provides a very helpful resource to gain intuition on how a neural network works.

In this tool, the MLP architectures are easily editable. It is possible to train the network with multiple hyperparameter combinations and obtain visual results throughout the training.

Vanilla neural networks, while powerful, have several drawbacks that motivate the exploration of alternative approaches such as physically-enhanced neural networks. A major limitation is their black-box nature, which makes it difficult to interpret the learned representations and understand the underlying mechanisms driving the predictions. In addition, vanilla neural networks often require large amounts of labeled data for training, which is not always readily available, especially in domains where data collection is expensive or time-consuming. Furthermore, traditional neural networks do not explicitly incorporate known physical principles or constraints into their architectures, which can lead to inefficiencies or inaccuracies when modeling physical systems or phenomena (Raissi et al., 2019). These limitations underscore the need for novel methods that aim to integrate domain-specific knowledge, such as physical laws or constraints, into the learning process to improve interpretability, robustness, generalization capabilities, and training efficiency, while harnessing the power of Deep Learning for complex prediction tasks.

2.2 Coupling techniques between physical knowledge and neural networks

The integration of Deep Learning with physical knowledge offers the possibility to improve the solution of complex scientific and engineering problems and is a growing trend, as shown in Figure 1.6. Due to the very large number of publications in this field, an exhaustive literature review is not possible. In Section 2.3, the contributions that deal specifically with neural networks for constitutive modeling are reviewed. Nevertheless, the techniques for incorporating physical knowledge into neural networks can be grouped into 3 families, which are described in detail below.

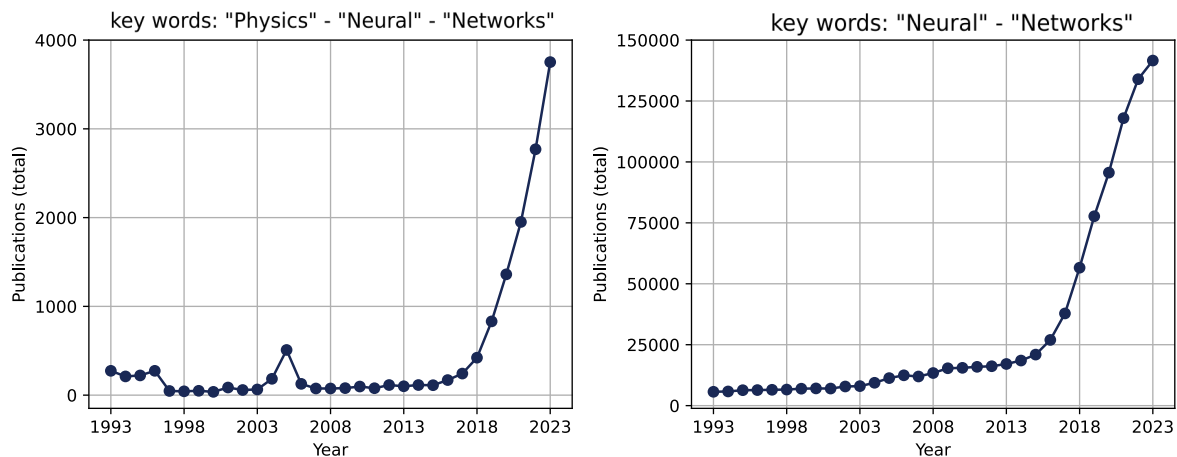


Figure 1.6 • Left: Number of publications per year containing the keywords "Physics", "Neural" and "Networks" in the title or abstract. Right: Number of publications per year containing the keywords "neural" and "networks" in the title or abstract. The database used is the one accessible from the website <https://app.dimensions.ai/discover/publication>. Other databases may give different results.

2.2.1 Physics-guided loss function

The integration of physical laws into the loss function of neural networks marks a significant advance in bridging the gap between Deep Learning and traditional scientific computing. Known as "physics-informed", this approach uses the governing physical equations, often partial differential equations (PDEs), as an important component of the training process. Unlike conventional neural networks, where the loss function primarily quantifies the deviation between predictions and data, this technique introduces an additional term to the loss function that informs the training with the underlying physical principles. Pioneering ideas for adding PDE residuals to the loss can be found in (Lagaris et al., 1997), while (Raissi et al., 2019) named the concept of "physics-informed neural networks" (PINN) and accelerated the physics community's enthusiasm for neural networks. PINNs are now used in many different applications, such as fluid mechanics (Cai et al., 2021), biology (Lagergren

et al., 2020), chemistry (Zanardi et al., 2022), magnetostatics (Kovacs et al., 2022), or solid mechanics (Haghighat et al., 2021), to name just a few.

The key to this methodology is the formulation of a PDE residual-based loss component. This component evaluates the extent to which the predictions of the neural network are consistent with the PDEs describing the physical system under study. Specifically, the network is designed to predict physical quantities of interest (e.g., temperature, pressure) over a domain, and the PDE residuals are computed by substituting these predictions into the corresponding PDEs. A well-trained PINN minimizes these residuals, ensuring that its predictions not only fit the available data, but also satisfy the physical laws governing the system.

Several elements are typically combined to construct the total loss function:

- **Data Loss:** This element measures the discrepancy between the neural network's predictions and the observed data, guiding the network to learn from empirical observations.
- **PDE Residual Loss:** This element, the hallmark of PINNs, quantifies the discrepancies between the predictions and the physical laws expressed by the PDEs. Minimizing this loss ensures that the solutions are physically plausible.
- **Boundary and Initial Condition Loss:** To further ensure the fidelity of the predictions, terms are included that penalize deviations from known boundaries and initial conditions of the physical system. This forces the neural network to obey these critical constraints.

The process of training a PINN involves the simultaneous minimization of these components of the loss function, using backpropagation and optimization techniques as in conventional Deep Learning. However, the inclusion of the PDE residual loss introduces a new dimension to the training process, requiring careful consideration of the physical context and the numerical stability of the learning algorithm, as well as a (potentially time-consuming) tuning of the weights between losses (Bischof & Kraus, 2022).

This approach of embedding physical knowledge directly into the loss function allows for the direct integration of physical constraints and principles, improving the reliability and accuracy of the neural network's predictions, especially in scenarios where empirical data may be sparse or noisy.

Remark

It is worth noting that the mCRE functional has similarities with the structure of the PINN loss function, with a decomposition that includes physical and data terms. However, the mCRE functional is not used to solve PDE, it is used to correct model bias. The minimization of the mCRE functional also involves strong enforcement of constraints which is not the case with PINN (penalization).

2.2.2 Physics-guided Architecture

A second way to couple physics and neural networks is to work at the level of the architecture of the neural network. A common approach is to construct network architectures that reflect the structure of the physical equations themselves. For example, networks can be designed to have layers or connections that correspond directly to terms in differential equations, ensuring that the flow of information through the network mimics the physical processes (Muralidhar et al., 2020). When dealing with a physical problem, even the choice of inputs and outputs of a neural network is often motivated by physical arguments. There are also hybrid approaches that combine neural networks with traditional numerical solvers for differential equations (Fang, 2022; Massala et al., 2023). In such architectures, the neural network may be responsible for learning complex boundary conditions or source terms, while the numerical solver handles the integration of the differential equations. This symbiosis makes it possible to leverage the strengths of established numerical methods while adding the flexibility and adaptability of Machine Learning models.

Another approach is to use specialized layers or units that enforce certain constraints. For example, symmetries and invariances inherent in the physical system can be embedded in the network design to ensure that the network's predictions remain consistent under transformations that leave the physical system unchanged (Cohen & Welling, 2016). Another example is a convolutional layer which naturally guarantees invariance under translation. Other properties, such as convexity, can also be enforced in the (Amos et al., 2017) architecture, as we will see.

The main advantage of enforcing physics in the architecture is that every prediction, in both the training and inference phases, satisfies the enforced properties, whereas non-respect of physical knowledge in the loss function is penalized only during training. Also, by constraining the neural network in the architecture, the search space (described by the neural networks) is reduced to only physical solutions, thus reducing the number of local minima in the training process.

2.2.3 Physics-guided initialization

Transfer learning is a transformative strategy in the field of neural networks, particularly for integrating physical knowledge into Machine Learning models (Farahani et al., 2020). In this approach, knowledge gained from solving one problem is applied to a different but related problem. In the context of embedding physical principles into neural networks, transfer learning offers a unique way to improve model performance, especially in scenarios where data is scarce or the complexity of physical systems is high (Heidenreich et al., 2024).

The essence of transfer learning in this domain involves pre-training a neural network on a vast dataset where the physical laws are well-understood and data are abundant. The network learns to recognize patterns and features governed by these underlying physical principles. This pre-trained model is then fine-tuned on a smaller, target dataset that represents a specific scientific or engineering problem of interest. The fine-tuning process adjusts the model's weights and biases to adapt to the nuances of the new problem, while

retaining the fundamental physical insights learned during pre-training.

Transfer learning significantly reduces the amount of data required to train a model on the target task. By using a model pre-trained on a related task, the network uses previously learned features and patterns, accelerating the learning process for the new task. To effectively implement transfer learning, it is critical to identify source tasks that are sufficiently related to the target task to ensure that the knowledge transfer is relevant and useful. In addition, careful attention must be paid to the fine-tuning process, balancing the retention of learned physical principles with adaptation to the specific characteristics of the new problem.

In summary, transfer learning is a powerful mechanism for integrating physical knowledge into neural networks. By using pre-trained models and adapting them to new tasks, this approach improves the efficiency, accuracy, and applicability of neural networks in solving complex scientific and engineering problems that are firmly grounded in physical laws.

2.3 Neural networks and constitutive modeling

The field of constitutive modeling also benefits from extensive research to evaluate the ability of neural networks to describe constitutive behavior. Although this topic is very trendy today, the idea of representing a constitutive model is not new, as the pioneering ideas can be found in (Ghaboussi et al., 1999). At that time, learning strain-stress relationships relied solely on data, neglecting any physical insight into the network architecture or loss function.

Recent contributions can be divided into two groups according to the data used for training. The first group of contributions aims at training neural networks in a supervised learning procedure with a strain-stress database (or strain-free energy), while the second group aims at discovering new constitutive models from data obtained from real experiments, thus training neural networks in an unsupervised way. The contributions of the first community focus more on architecture, while the second community focuses more on numerical methods suitable for training neural networks with real data (or at least synthetic data representing obtainable measures). The main focus of this manuscript belongs to the second group of contributions, but is closely related to the first. The supervised contributions can be motivated by different reasons:

- **Speed-up simulation:** Once a neural network is trained, the inference phase of a constitutive artificial neural network is generally much faster compared to other traditional numerical solvers in computational mechanics. Trained with a strain-stress database, a neural network can easily replace the complex integration of evolution laws (Rezaei et al., 2024) or be used as a homogenization tool replacing complex multiscale simulation (Bishara et al., 2023).
- **Gain knowledge about neural networks for constitutive modeling to pave the way for model discovery:** Another motivation for developing supervised learning of constitutive models is simply to improve the understanding of neural networks for constitutive modeling or to propose new modeling frameworks.

In the following paragraphs, some choices are discussed, such as what to put in the inputs and outputs of the neural network, the different thermodynamic frameworks, the types of architectures used, how to handle internal variables, or the choice of loss functions. In fact, the following discussion of these choices is nothing more than a discussion of modeling choices. Some of these questions predate the neural network trend and may be answered differently in different contexts.

Remark

At the beginning of this PhD in 2021, most of the work on constitutive architecture was just beginning. Some approaches that now seem well established in 2024 were not published at all at the beginning of the PhD. In the following literature review, precision is added when contributions share ideas developed simultaneously in this PhD.

On the choice of inputs and outputs

When using neural networks to model material behavior, the choice of input and output variables is critical.

This question first addresses the scale at which the modeling is aimed. Global information about the structure can be used to predict whether the structure is damaged or not (Gulgec et al., 2017; Bono et al., 2023), as a damage detection classification problem (damage or not damage). On the other hand, models can incorporate input information related to the microstructure of the material (Aldakheel et al., 2023; Rao & Liu, 2020). In between these two extremes are mesoscopic models based on the thermodynamics of continuous media with a description based on internal variables. This manuscript focuses on the latter scale.

Once the modeling scale is determined, it can be argued that modeling all of the constitutive behavior may be overkill, and the neural network should be used only to represent the unknown (or difficult to model) part of the behavior. Learning linear elasticity with a neural network may be a questionable idea. For example, in the context of plasticity and motivated by modeling reasons, (Liu et al., 2022) uses the neural network only to learn the yield surface. In the same philosophy, (Fuhg et al., 2023) proposes a modular framework for elastoplasticity in which neural networks only represent the unknown physics. This kind of idea is particularly suitable in a low-data regime, which is generally the case in experimental setups.

Finally, to go further on the choice of inputs in a constitutive neural network, emphasis should be placed on the invariant formulation of constitutive models (Boehler, 1987). If the class of symmetry of the material to be modeled is known, using the associated invariants of strain in the input (instead of strain components) allows to write a symmetry preserving model. Encoding symmetry in this way reduces the amount of data needed to train the networks, which is one explanation for the recent popularity of this choice (Linden et al., 2023; Linka & Kuhl, 2023; Fuhg et al., 2022; Tac et al., 2022; Thakolkaran et al., 2022). Some of these papers have been published simultaneously with some studies of this thesis.

On the choice of thermodynamics framework

It is well known that constitutive models must satisfy thermodynamics. The respect of thermodynamics can be learned from data, but one can argue that a neural network would need a very large amount of data to derive thermodynamic principles on its own. Furthermore, learning thermodynamics from data does not give any guarantee on its respect in the inference phase.

The question then arises on how to ensure that the network makes only thermodynamically consistent predictions.

Several works, including studies from this thesis, have addressed the issue of enforcing thermodynamic constraints in the architecture or penalizing their violation in the loss function during the training phase. The advantage of enforcing constraints in the architecture is that they are automatically satisfied in the inference phase, whereas this is not the case when constraints are imposed in the loss function during training. The first coupling between thermodynamics knowledge and neural network (in order of appearance) is called "thermodynamic artificial neural network" (TANN) and was introduced in a series of works by Masi. In (Masi & Stefanou, 2022) a multilayer perceptron is proposed to predict the stress increment derived from a thermodynamic potential. The mechanical dissipation is then computed and its positivity constraint is included in the loss function used for training. Nevertheless, this remains only a penalization during the training phase, which does not guarantee anything concerning the inference. The dissipation can also be computed in the inference phase to check whether the Clausius-Duhem inequality is satisfied or not. In (He & Chen, 2022) the stress is also derived from a thermodynamic potential, but the increment of the internal variable is given by a recurrent neural network. The main advantage is that this approach does not require the selection of the internal variables as they are obtained from the internal memories of the recurrent cells. The second principle is also considered in the loss function.

Another way to define a thermodynamically consistent architecture can be found in (Hernández et al., 2021), where the metriplectic structure of dissipative systems is implemented in the form of the so-called General Equation for Non-Equilibrium Reversible-Irreversible Coupling (GENERIC) (Romero, 2009; Öttinger & Grmela, 1997). In (Fuhg et al., 2023) thermodynamic potentials are predicted with the input-convex architecture proposed in (Amos et al., 2017) (with internal variables as input), thus automatically satisfying the Clausius-Duhem inequality. Finally, the Generalized Standard Material framework (Halphen & Nguyen, 1975) is constrained using input-convex neural networks (Amos et al., 2017; Rosenkranz et al., 2024), as is done in this manuscript. Note that the GENERIC and Generalized Standard Material frameworks share similarities and the transition from one to the other can be found in (Mielke, 2011) as in both frameworks laws are derived from potentials. The GENERIC framework can represent more than just constitutive modeling (e.g. dynamical systems), but when this framework is used to represent constitutive laws, it is equivalent to the Generalized Standard Material framework.

In this manuscript, the Generalized Standard Material framework is used, which means that convex potentials must be constructed. The following details how such requirements

can be enforced in the architecture:

The architecture used is shown in Figure 1.7. To ensure convexity, the weights between the layers are positive (the weights mapping the input layer to the layers can be negative), and the activation functions are convex and non-decreasing. In the case without evolution laws, this framework is also widely used, with only one potential: the Helmholtz free energy (Linden et al., 2023; Linka & Kuhl, 2023; Fuhg et al., 2022; Tac et al., 2022; Thakolkaran et al., 2022; As'ad et al., 2022). The stress is obtained by automatic differentiation of the free energy:

$$\boldsymbol{\sigma} = \frac{\partial \psi}{\partial \boldsymbol{\epsilon}} \quad (1.39)$$

$\psi(\boldsymbol{\epsilon})$ and $\boldsymbol{\sigma} = \frac{\partial \psi}{\partial \boldsymbol{\epsilon}}$ should vanish at zero deformation ($\boldsymbol{\epsilon} = \mathbf{0}$) so the potential is written:

$$\psi(\boldsymbol{\epsilon}) = \psi_{NN}(\boldsymbol{\epsilon}) - \psi_{NN}(\boldsymbol{\epsilon} = \mathbf{0}) - \left. \frac{\partial \psi_{NN}}{\partial \boldsymbol{\epsilon}} \right|_{\boldsymbol{\epsilon}=\mathbf{0}} : \boldsymbol{\epsilon} \quad (1.40)$$

where ψ_{NN} is the output of the neural network.

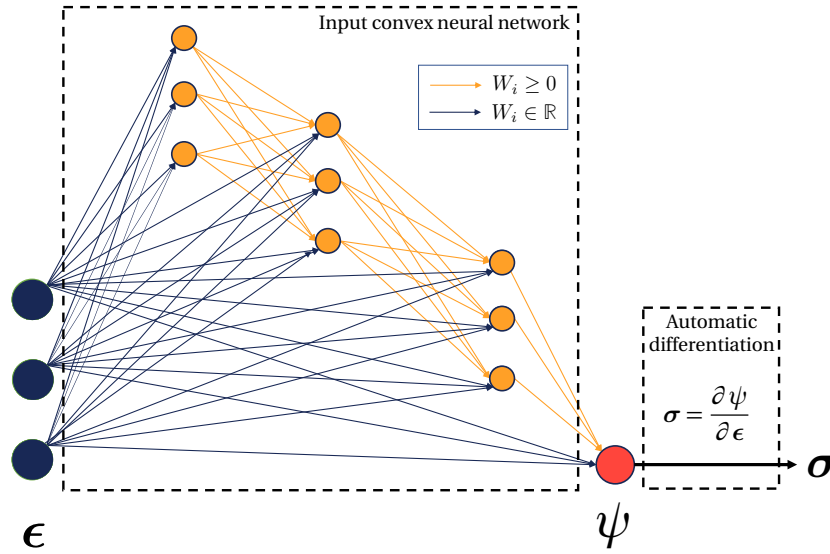


Figure 1.7 • Input convex thermodynamics-consistent architecture used.

On the role of internal variables

The role of internal variables is also very important. Internal variables characterize the state of matter. They are hidden variables in the sense that they are not measurable. Different approaches to the treatment of internal variables can be found in the literature. The choices made in the literature are closely related to the amount of information available. Indeed, with rich information, it is quite possible to detect internal variables with a neural network. However, in contexts with limited data availability, learning internal variables becomes very complicated and they have to be postulated a priori (Fuhg et al., 2023). To my knowledge, there are no studies that have succeeded in automatically learning internal

variables based on truly measurable information. There are two interesting approaches to learning internal variables. On the one hand, recurrent networks can be adapted to learn internal variables thanks to their memory system (Rosenkranz et al., 2024). In (Bonatti & Mohr, 2021), the network manages to learn a set of internal variables with a memory system inspired by LSTM. Since the data used to train the model are generated for a known model, this study showed that the number of learned internal variables can be relevant compared to the physical model. On the other hand, autoencoders, an architecture useful for compressing information, are also suitable for learning internal variables. In (Masi & Stefanou, 2022), an autoencoder system is used for homogenization to reconstruct microstructural information while constraining the dimension of the latent space.

Type of NN structure

The choice of network architecture is closely related to the choice of thermodynamic framework and internal variables. A large number of works have addressed the issue of representing history-dependent constitutive models with neural networks (Wu et al., 2020; Zhang & Mohr, 2020; Bonatti & Mohr, 2022; Gorji et al., 2020; Vlassis & Sun, 2021). As is classically the case in Deep Learning, different types of neural networks are used, such as

- feed-forward neural networks. With this type of network, the inputs of the neural network are generally the strain increment and the postulated internal variables (see for examples (Ibragimova et al., 2021; Weber et al., 2023));
- recurrent neural networks which are suited to deal with time series. One strong interest of such recurrent networks is the use of internal memories which can play the role of internal variables (see for examples (Bonatti & Mohr, 2021; Im et al., 2021; Wu et al., 2020; Mozaffar et al., 2019; Gorji et al., 2020));
- time convolutional neural networks that are also suited for time series: a comparison between time convolutional neural networks and recurrent neural networks can be found in (Abueidda et al., 2021) in the context of plasticity and thermo-viscoplasticity.

In (Linka & Kuhl, 2023), an architecture specifically designed for constitutive modeling is proposed, in which functions usually encountered in constitutive models are embedded in the network architecture.

On the choice of loss function

Finally, the choice of loss function for training is an important issue. If a model is already available, it is possible to generate databases of strain-stress (or potential-strains), and in this case the choice of a least squares loss function is quite natural. In the supervised case, a discussion may arise as to whether it is better to place a loss function on the stress or on the potential (Vlassis & Sun, 2021). The answer to this question may vary depending on the way the network is parameterized.

In the unsupervised case, the task of choosing the loss function is more challenging. The choice of such a metric is not new to neural networks, but is a recurring question in inverse problems. Indeed, in the case of inverse problems for the identification of constitutive law parameters (Bonnet & Constantinescu, 2005), there are numerous identification

methods which can be based on least squares minimization such as the Finite Element Model Updating (FEMU) (Grédiac, 2004), on stochastic approaches (Kaipio & Somersalo, 2007; Tarantola, 2005; Rosic et al., 2013), or minimization of a cost function with a physical sense such as the Equilibrium Gap Method (Claire et al., 2004).

The Equilibrium Gap Method has been adapted to a behavior described by a neural network in the so-called NN-EUCLID framework (Thakolkaran et al., 2022), while the adaptation of the mCRE is the main goal of this thesis.

Chapter conclusion

This bibliographic chapter served to recall the chosen framework for inverse problems: the mCRE. This chapter also reviewed the basics of Deep Learning, with particular emphasis on the coupling of techniques with physical knowledge, especially the use of neural networks for learning constitutive laws. Several lessons can be drawn from this bibliography for this manuscript. First, the literature review on mCRE has identified the current limitations of work on parameter identification using this tool. A notable point is that despite the widespread use of the mCRE, relatively few studies have focused on updating parameters for nonlinear constitutive models. It is also important to note that all works on the mCRE assume a model form and attempt to identify the parameters of that model. The model form has always been considered reliable, while the parameters have been considered unreliable. This manuscript then deals with how to free the model form and considers the model form to be unreliable as well.

The second set of lessons comes from the bibliographic study of coupling techniques between Deep Learning and physical knowledge. Recent work has addressed many points of resistance to the use of neural networks. Indeed, it is now possible to constrain the neural network so that it does not violate physics, for example by enforcing a thermodynamic framework or symmetry properties. These works thus point to the possibility of using neural networks in critical contexts, such as the one targeted in this manuscript. This study also highlights the lack of work dealing with unsupervised learning, which is required to train neural networks with measurable data. The remainder of this manuscript then focuses on the unsupervised learning of constitutive laws described by a neural network that validates physical properties within the mCRE framework, in a unified strategy.

2

Learning nonlinear state laws with neural networks trained in the mCRE framework

The work presented in this chapter has been the subject of the following publication: Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. NN-mCRE: a modified Constitutive Relation Error framework for unsupervised learning of nonlinear state laws with physics-augmented Neural Networks. *International Journal for Numerical Methods in Engineering*, 2024, 125(8):e7439. <https://dx.doi.org/10.1002/nme.7439>

Contents

| | | |
|----------|--|-----------|
| 1 | Problem definition | 40 |
| 2 | A minimization procedure suited for nonlinear elasticity | 41 |
| 2.1 | Step 1 | 42 |
| 2.2 | Step 2 | 44 |
| 3 | Automatic rules for hyperparameters tuning | 45 |
| 3.1 | Reducing the sensitivity to initialization: physics-guided initialization with <i>a priori</i> knowledge | 46 |
| 3.2 | Reducing the sensitivity to weighting between losses: automatic tuning of α | 46 |
| 3.3 | Reducing the sensitivity to learning rate: automatic tuning of learning rate | 47 |
| 3.4 | Reducing the sensitivity to the number of epochs: definition of a physics-based stop criterion | 49 |
| 3.5 | Extension to the case of multiple loadings in the database | 50 |
| 4 | Practical implementation | 50 |
| 5 | Results | 51 |

| | | |
|-------|---|----|
| 5.1 | Application on a nonlinear case in small deformation | 51 |
| 5.1.1 | Training database | 51 |
| 5.1.2 | Validation database | 52 |
| 5.1.3 | Error after initialization training | 52 |
| 5.1.4 | Evaluation of the method | 53 |
| 5.2 | Application on a hyperelastic case: Mooney-Rivlin model | 54 |
| 5.2.1 | Training database | 55 |
| 5.2.2 | Validation database | 56 |
| 5.2.3 | Evaluation of the method, case of incomplete training database | 57 |
| 5.3 | Robustness on hyperparameters choice | 57 |
| 5.3.1 | Relevance of adaptive learning rate | 58 |
| 5.3.2 | Relevance of adaptive weighting between losses | 58 |
| 5.3.3 | Robustness to noise | 59 |

The previous bibliography chapter suggested the possibility of unsupervised training of a physically constrained neural network within the mCRE framework. This chapter aims to explore the first methodological aspects related to the coupling of Deep Learning with the mCRE framework. In order to gradually consider the difficulties, this chapter focuses on a nonlinear, but history-independent, behavior. After a brief description of the problem addressed in this chapter, a minimization procedure adapted to nonlinear elastic behavior is presented. This procedure is a continuation of the work on mCRE with alternating minimization. The free energy is represented by the mean of an input-convex neural network, and the goal of the learning is to find the neural network parameters that best approximate the constitutive model. The training of neural networks is much more challenging than classical parameter identification due to the very large number of parameters to be learned. This difficulty is aggravated by the fact that the numerical training method requires the tuning of many hyperparameters, often manually adjusted by a user. Unfortunately, the context of DDDAS targeted in this manuscript prevents relying on a user to adjust the hyperparameters. Therefore, special attention is paid to the automatic tuning of hyperparameters. Finally, the method is validated on two test cases. The first case involves the learning of a nonlinear elastic law under the hypothesis of small perturbations, and the second case is a hyperelastic case involving the learning of a Mooney-Rivlin model.

1 Problem definition

To define the problem notations, let us consider a body in initial configuration $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$) with boundary $\partial\Omega$ and isothermal environment. Dirichlet boundary conditions are imposed on $\partial\Omega_1 \subset \partial\Omega$ by means of a displacement field \mathbf{u}_d . Neumann boundary conditions are prescribed on $\partial\Omega_2 \subset \partial\Omega$ by means of a force field \mathbf{f}_s^d , with $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$. A body force

field \mathbf{f}_v^d may also be prescribed in Ω . Additionally, strain measurements \mathbf{E}_{obs} (in the case of observations from optical fibers or strain gauges) or displacement measurements \mathbf{u}_{obs} (in the case of digital image correlation) are available. We consider here observation data \mathbf{E}_{obs} .

The solution to the mechanical problem is the couple (\mathbf{u}, \mathbf{S}) of displacement and second Piola-Kirchoff stress fields that satisfies the three following groups of equations:

- **kinematic admissibility** defines the space \mathcal{U}_{ad} of displacement fields satisfying the Dirichlet boundary conditions:

$$\mathbf{u}|_{\partial\Omega_1} = \mathbf{u}_d$$

- **static admissibility** defines the space \mathcal{S}_{ad} of stress fields satisfying the equilibrium:

$$\int_{\Omega} \mathbf{S}(\mathbf{u}) : (\mathbf{F}(\mathbf{u})^T \cdot \nabla \mathbf{v}) d\Omega = \int_{\Omega} \mathbf{f}_d^v \cdot \mathbf{v} d\Omega + \int_{\partial\Omega_2} \mathbf{f}_d^s \cdot \mathbf{v} dS \quad \forall \mathbf{v} \in \mathcal{U}_0$$

with \mathbf{F} the deformation gradient and \mathcal{U}_0 the space of kinematic admissibility with homogeneous Dirichlet conditions.

- **constitutive relation:**

$$\mathbf{S} = \frac{\partial \psi(\mathbf{E}; \mathbf{p})}{\partial \mathbf{E}}$$

with the strain tensor $\mathbf{E}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \nabla \mathbf{u})$ and \mathbf{p} the parameters of the constitutive law. Such a notation for the constitutive law, involving a convex potential ψ depending on state variables alone, is provided by the first principle of thermodynamics.

The minimization of the mCRE aims to identify the constitutive relation parameters \mathbf{p} (weight and bias of a neural network) that fit the best to experimental data. Here Dirichlet and Neumann's conditions are known, as well as the observations \mathbf{E}_{obs} or \mathbf{u}_{obs} that are affected by measurement noise.

2 A minimization procedure suited for nonlinear elasticity

In the present context, the minimization of the mCRE aims to find the parameters \mathbf{p}_{opt} such that:

$$\mathbf{p}_{opt} = \operatorname{argmin}_{\mathbf{p} \in \mathcal{P}} \left[\min_{(\mathbf{u}, \mathbf{S}) \in \mathcal{A}_d} \left[\int_{\Omega} (\psi(\hat{\mathbf{u}}; \mathbf{p}) + \psi^*(\hat{\mathbf{S}}; \mathbf{p}) - \hat{\mathbf{S}} : \mathbf{E}(\hat{\mathbf{u}})) d\Omega + \alpha \|\Pi \mathbf{E}(\hat{\mathbf{u}}) - \mathbf{E}_{obs}\|^2 \right] \right] \quad (2.1)$$

where the potential ψ and the space of parameters \mathcal{P} are such that the constitutive model guarantees the physical conditions detailed in the paragraph dedicated to the thermodynamic framework in Section 2.3. This section describes the minimization in a unified framework suited for:

- an explicitly given parametrized constitutive relation.

- a constitutive relation described by a neural network (in this case, parameters \mathbf{p} should be understood as weights and biases of the network). The description of a constitutive relation by a physics-augmented neural network enables to search for the potential ψ in a space of potentials that satisfy the requirements of a constitutive law. Here, input-convex neural networks (such as detailed in Section 2.3 Figure 1.7) are used to represent the potential ψ . Discussions on the network architecture will be conducted in Chapter 4.

This minimization consists in an iterative process where iteration $n + 1$ is composed of:

- a first step performed with fixed parameters $\mathbf{p}^{(n)}$, in which a new admissible field couple $(\hat{\mathbf{u}}, \hat{\mathbf{S}})^{(n+1)}$ is found to minimize the mCRE.
- a second step performed with fixed admissible field couple $(\hat{\mathbf{u}}, \hat{\mathbf{S}})^{(n+1)}$, in which parameters are updated with a gradient descent step.

Both steps are detailed in the following section for iteration n . A novelty of this approach lies in the way the Step 1 is performed: in the present context this Step 1 is nonlinear and the minimization is performed with a Newton scheme.

2.1 Step 1

In this step, parameters $\mathbf{p}^{(n)}$ are fixed and the goal is to find:

$$(\hat{\mathbf{u}}, \hat{\mathbf{S}})^{(n+1)} = \underset{(\hat{\mathbf{u}}, \hat{\mathbf{S}}) \in \mathcal{A}_d}{\operatorname{argmin}} \left[\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{S}}; \mathbf{p}^{(n)}) \right] \quad (2.2)$$

with

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{S}}; \mathbf{p}^{(n)}) = \int_{\Omega} \left(\psi(\hat{\mathbf{u}}; \mathbf{p}^{(n)}) + \psi^*(\hat{\mathbf{S}}; \mathbf{p}^{(n)}) - \hat{\mathbf{S}} : \mathbf{E}(\hat{\mathbf{u}}) \right) d\Omega + \alpha \|\Pi \mathbf{E}(\hat{\mathbf{u}}) - \mathbf{E}_{obs}\|^2 \quad (2.3)$$

To express the Legendre-Fenchel transform $\psi^*(\hat{\mathbf{S}})$ (defined in Section 1.5.1) in a more convenient way, a displacement field $\hat{\mathbf{v}}$ is introduced. With the convexity of ψ with respect to \mathbf{E} the sup is reached for $\hat{\mathbf{v}}$ such that:

$$\hat{\mathbf{S}} = \left. \frac{\partial \psi}{\partial \mathbf{E}} \right|_{\mathbf{E}(\hat{\mathbf{v}})} \quad (2.4)$$

so that,

$$\psi^*(\hat{\mathbf{S}}) = \left. \frac{\partial \psi}{\partial \mathbf{E}} \right|_{\mathbf{E}(\hat{\mathbf{v}})} : \mathbf{E}(\hat{\mathbf{v}}) - \psi(\mathbf{E}(\hat{\mathbf{v}})) \quad (2.5)$$

By replacing in (2.3) the mCRE reads:

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \mathbf{p}) = \int_{\Omega} \left(\psi(\hat{\mathbf{u}}; \mathbf{p}^{(n)}) - \psi(\hat{\mathbf{v}}; \mathbf{p}^{(n)}) + \left. \frac{\partial \psi}{\partial \mathbf{E}} \right|_{\mathbf{E}(\hat{\mathbf{v}})} : (\mathbf{E}(\hat{\mathbf{v}}) - \mathbf{E}(\hat{\mathbf{u}})) \right) d\Omega + \alpha \|\Pi \mathbf{E}(\hat{\mathbf{u}}) - \mathbf{E}_{obs}\|^2 \quad (2.6)$$

To search the couple $(\hat{\mathbf{u}}, \hat{\mathbf{S}})^{(n+1)}$ in the admissible space \mathcal{A}_d :

- $\hat{\mathbf{u}}$ is discretized with the following decomposition:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_a \\ \mathbf{U}_d \end{bmatrix} \quad (2.7)$$

where \mathbf{U}_a is the discretized displacement field at active degrees of freedom and \mathbf{U}_d the discretized displacement field for the Dirichlet boundary conditions.

- the mCRE minimization is performed under the constraint:

$$h(\mathbf{v}, \boldsymbol{\lambda}) = \int_{\Omega} \mathbf{S}(\mathbf{v}) : (\mathbf{F}(\mathbf{v})^T \cdot \nabla \boldsymbol{\lambda}) \, d\Omega - \int_{\Omega} \mathbf{f}_d^v \cdot \boldsymbol{\lambda} \, d\Omega - \int_{\partial\Omega_2} \mathbf{f}_d^s \cdot \boldsymbol{\lambda} \, dS = \mathbf{0} \quad \forall \boldsymbol{\lambda} \in \mathcal{U}_0 \quad (2.8)$$

so that $\hat{\mathbf{S}}$ satisfies the static admissibility.

This constrained minimization is performed by means of a Lagrangian functional:

$$\mathcal{L}(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \boldsymbol{\lambda}) = \mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{v}}) + h(\hat{\mathbf{v}}, \boldsymbol{\lambda}) \quad (2.9)$$

with the following Newton scheme (Boyd & Vandenberghe, 2004):

$$\begin{bmatrix} \hat{\mathbf{u}}_{k+1} - \hat{\mathbf{u}}_k \\ \hat{\mathbf{v}}_{k+1} - \hat{\mathbf{v}}_k \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{u}}^2} & \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{u}} \partial \hat{\mathbf{v}}} & \mathbf{0} \\ \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{v}} \partial \hat{\mathbf{u}}} & \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{v}}^2} & \frac{\partial h(\hat{\mathbf{v}}, \boldsymbol{\lambda})}{\partial \hat{\mathbf{v}}} \\ \mathbf{0} & \frac{\partial h(\hat{\mathbf{v}}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{u}}} \\ \frac{\partial \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{v}}} \\ h(\hat{\mathbf{v}}, \boldsymbol{\lambda}) \end{bmatrix} \quad (2.10)$$

The right-hand side of (2.10) is evaluated with the quantities of iteration k .

The first-order derivatives are:

$$\begin{cases} \frac{\partial \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{u}}} = \int_{\Omega} \left(\frac{\partial \mathbf{E}}{\partial \hat{\mathbf{u}}} \left(\frac{\partial \psi}{\partial \mathbf{E}} \Big|_{\boldsymbol{\epsilon}(\hat{\mathbf{u}})} - \frac{\partial \psi}{\partial \mathbf{E}} \Big|_{\boldsymbol{\epsilon}(\hat{\mathbf{v}})} \right) \right) d\Omega + \alpha \frac{\partial \mathbf{E}}{\partial \hat{\mathbf{u}}} (\Pi \mathbf{E}(\hat{\mathbf{u}}) - \mathbf{E}_{obs}) \\ \frac{\partial \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{v}}} = \int_{\Omega} \frac{\partial \mathbf{E}}{\partial \hat{\mathbf{v}}} \frac{\partial^2 \psi}{\partial \mathbf{E}^2} \Big|_{\mathbf{E}(\hat{\mathbf{v}})} (\mathbf{E}(\hat{\mathbf{v}}) - \mathbf{E}(\hat{\mathbf{u}})) \, d\Omega \\ h(\hat{\mathbf{v}}, \boldsymbol{\lambda}) = - \int_{\Omega} \left(\frac{\partial \psi}{\partial \mathbf{E}} \Big|_{\mathbf{E}(\hat{\mathbf{v}})} : \mathbf{F}(\hat{\mathbf{v}}) \nabla \boldsymbol{\lambda} - \mathbf{f}_d^v \boldsymbol{\lambda} \right) d\Omega - \int_{\partial_2 \Omega} \mathbf{f}_d^s \boldsymbol{\lambda} \, dS \end{cases} \quad (2.11)$$

Concerning second-order derivatives, terms that vanish at convergence of the Newton

scheme are neglected so that:

$$\begin{cases} \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{u}}^2} = \int_{\Omega} \left(\left(\frac{\partial \mathbf{E}}{\partial \hat{\mathbf{u}}} \right)^2 \frac{\partial^2 \psi}{\partial \mathbf{E}^2} \Big|_{\mathbf{E}(\hat{\mathbf{u}})} \right) d\Omega + \alpha \left(\frac{\partial \boldsymbol{\epsilon}}{\partial \hat{\mathbf{u}}} \right)^2 \Pi \cdot \Pi^T \\ \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{v}}^2} = \int_{\Omega} \left(\frac{\partial \mathbf{E}}{\partial \hat{\mathbf{v}}} \right)^2 \frac{\partial^2 \psi}{\partial \mathbf{E}^2} \Big|_{\mathbf{E}(\hat{\mathbf{v}})} d\Omega \\ \frac{\partial^2 \mathcal{E}_{mCRE}^2}{\partial \hat{\mathbf{u}} \partial \hat{\mathbf{v}}} = \int_{\Omega} - \frac{\partial \mathbf{E}}{\partial \hat{\mathbf{v}}} \frac{\partial \mathbf{E}}{\partial \hat{\mathbf{u}}} \frac{\partial^2 \psi}{\partial \mathbf{E}^2} \Big|_{\mathbf{E}(\hat{\mathbf{v}})} d\Omega \\ \frac{\partial h(\hat{\mathbf{v}}, \boldsymbol{\lambda})}{\partial \hat{\mathbf{v}}} = \int_{\Omega} \left(\frac{\partial \mathbf{E}}{\partial \hat{\mathbf{v}}} \frac{\partial^2 \psi}{\partial \mathbf{E}^2} \Big|_{\mathbf{E}} \mathbf{F}(\hat{\mathbf{v}}) \nabla \boldsymbol{\lambda} + \frac{\partial \psi}{\partial \mathbf{E}} \Big|_{\mathbf{E}(\hat{\mathbf{v}})} \frac{\partial \mathbf{F}(\hat{\mathbf{v}})}{\partial \hat{\mathbf{v}}} \nabla \boldsymbol{\lambda} \right) d\Omega \end{cases} \quad (2.12)$$

with $\mathbf{F}(\hat{\mathbf{v}}) = (1 + \nabla \hat{\mathbf{v}})$.

All the integrals are performed using finite element discretization (1.3 shows the discretization in the case of linear elasticity). The initialization of the Newton scheme is done with the fields $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ obtained at the end of Step 1 with the previous parameters. Step 1 terminates when the relative update on the displacement field is below a tolerance (usually set to 10^{-4}).

2.2 Step 2

This step consists of the updating of the parameters \mathbf{p} with a gradient descent step:

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} - l_r \frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}^{(n+1)}, \hat{\mathbf{v}}^{(n+1)}; \mathbf{p}^{(n)})}{d\mathbf{p}} \quad (2.13)$$

with $(\hat{\mathbf{u}}^{(n+1)}, \hat{\mathbf{v}}^{(n+1)})$ the fields obtained at the end of Step 1 of the $n + 1$ iteration of mCRE minimization. For the sake of simplicity, $(\hat{\mathbf{u}}^{(n+1)}, \hat{\mathbf{v}}^{(n+1)})$ are denoted $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ in the following. The gradient of mCRE with respect to the parameters is computed with the adjoint-state method:

$$\frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \mathbf{p})}{d\mathbf{p}} = \frac{d\mathcal{L}(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \boldsymbol{\lambda}; \mathbf{p})}{d\mathbf{p}} = \frac{d\hat{\mathbf{u}}}{d\mathbf{p}} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{u}}} + \frac{d\hat{\mathbf{v}}}{d\mathbf{p}} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{v}}} + \frac{d\boldsymbol{\lambda}}{d\mathbf{p}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} + \frac{\partial \mathcal{L}}{\partial \mathbf{p}} = \frac{\partial \mathcal{L}}{\partial \mathbf{p}} \quad (2.14)$$

$$\frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \mathbf{p})}{d\mathbf{p}} = \int_{\Omega} \left(\frac{\partial \psi}{\partial \mathbf{p}} \Big|_{\mathbf{E}(\hat{\mathbf{u}})} - \frac{\partial \psi}{\partial \mathbf{p}} \Big|_{\mathbf{E}(\hat{\mathbf{v}})} \right) d\Omega \quad (2.15)$$

with \mathcal{L} the Lagrangian associated to the constrained minimization problem of Step 1 defined in Section 1.1.1 (minimization defined in (2.2) with the constraint defined in (2.8)):

It is worth noticing in (2.15) that the parameters are updated following the computation of the gradient in the whole structure and not only where there is a measurement.

To summarize this section and introduce the next one, Figure 3.2 illustrates the general methodology for further training the physics-augmented neural network with the mCRE.

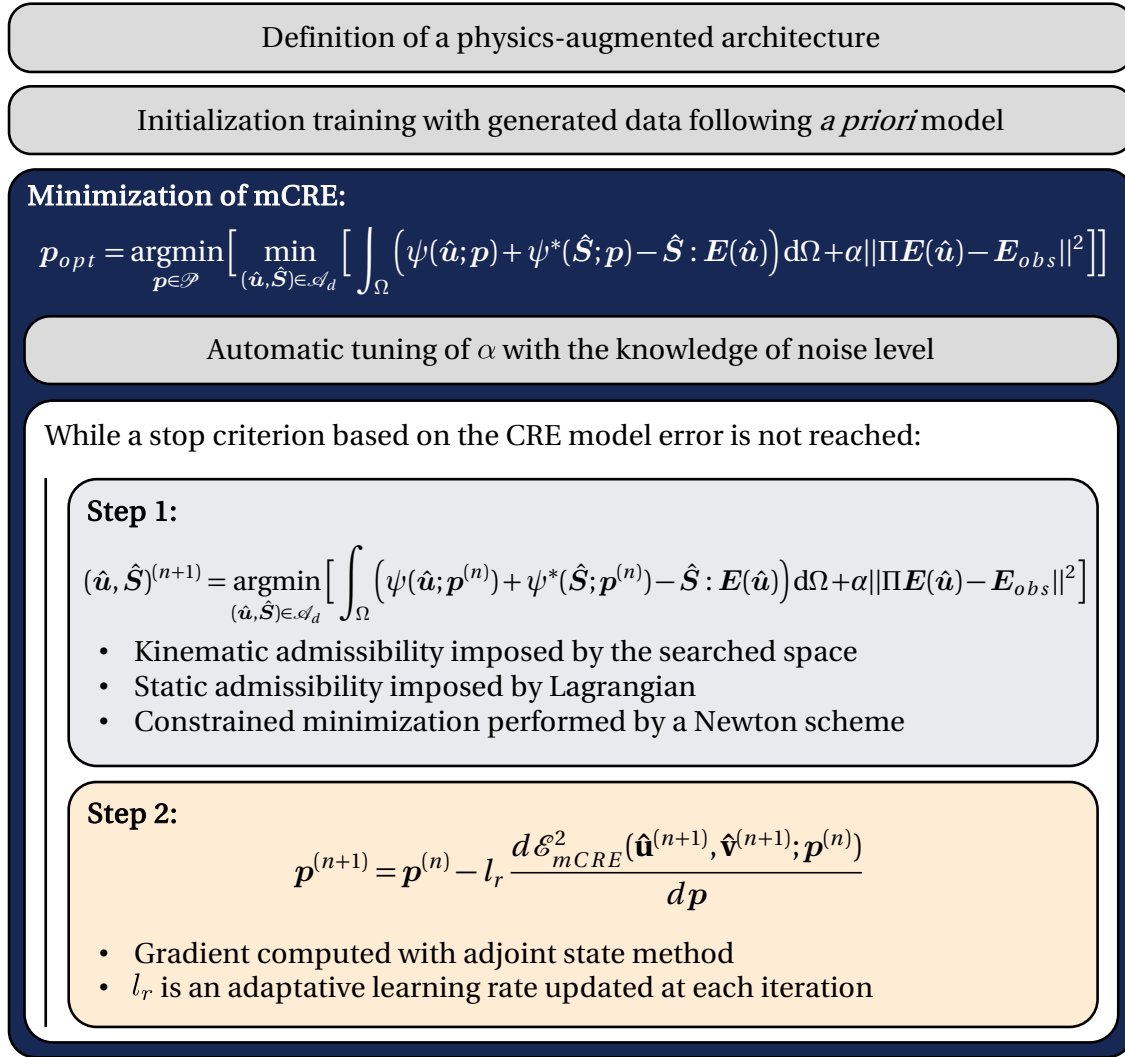


Figure 2.1 • Description of the method developed.

3 Automatic rules for hyperparameters tuning

When the constitutive model is described by a neural network, the minimization task is much harder because of the large number of parameters. Indeed, neural networks are known to be hard to train (Glorot & Bengio, 2010), partly because of the sensitivity to user-defined hyperparameters. Moreover, the DDDAS framework prevents the user from changing hyperparameters and performing several times the training. In the following, this issue is tackled by:

- a physics-guided initialization described in Section 3.1;
- a Morozov-based automatic tuning of the weighting between losses introduced in Section 3.2;
- an empirical adaptive learning rate rule described in Section 3.3;
- a CRE-based stop criterion used to avoid the predefined choice of the number of epochs (Section 3.4).

3.1 Reducing the sensitivity to initialization: physics-guided initialization with *a priori* knowledge

The training of neural networks is known to be sensitive to initialization (Glorot & Bengio, 2010), so is the minimization of the mCRE in the case of parameter updating (Nguyen, 2021). Indeed, with random initialization, the mCRE minimization fails to converge in most of the experiments performed. A common approach is to initialize the neural networks randomly following different distributions such as the one used in (Glorot & Bengio, 2010) (known as "Xavier uniform" and "Xavier normal"), in (He et al., 2015) (known as Kaiming initialization) or in (Saxe et al., 2014). These random initiations are sometimes associated with a normalization of input data in order to avoid exploding and vanishing gradients. In the context of neural networks for constitutive modeling, some approaches even train several neural networks with different initializations and choose the best after training (Thakolkaran et al., 2022).

Yet, when dealing with constitutive modeling an initial idea can be available such as is the case in this work. Therefore, before training within the mCRE framework, the network is initialized following an initial guess. Instead of random initialization, a first training is performed to initialize the network with an *a priori* model with the expected material parameters. A database $\{\tilde{\epsilon}, \tilde{\psi}\}$ is generated from a known constitutive model assumed to be close to the target one. The initialization training is performed in a classical supervised way with the Adam optimizer (Kingma & Ba, 2015) in order to minimize the mean square error loss. The positivity constraint on the weights is already enforced through gradient clipping. The choice of the *a priori* model can depend on the structure and type of loading considered.

The generated database is created to be rich enough to describe the studied range. The weights and biases of the neural network obtained at the end of the training are used as the initialization of the mCRE minimization algorithm described in the following section.

3.2 Reducing the sensitivity to weighting between losses: automatic tuning of α

A frequent concern in multiple losses optimization is the tuning of the weighting between losses (Bischof & Kraus, 2022; Wang et al., 2020). This concern is also known in the case of mCRE minimization, with the parameter α (Deraemaeker et al., 2004; Diaz et al., 2022). Here, the physical sense of the mCRE can provide a way to tune α : α is automatically tuned before the training and constantly adapted during the training. The general philosophy of the tuning is that the model should not be updated below the noise level, considering the noise level of the measurement tools is known a priori. Hence, the field $\hat{\mathbf{u}}$ constructed at the end of the Newton scheme (2.10) in Step 1 should get as close as possible to the measure, but should not fit the noise. To quantify this criterion, α is re-written:

$$\alpha = \alpha' \frac{1}{n_{obs} \sigma^2}$$

where n_{obs} is the number of observations and σ the standard deviation of the measurement noise. Replacing in (2.2):

$$(\hat{\mathbf{u}}, \hat{\mathbf{S}})^{(n+1)} = \underset{(\hat{\mathbf{u}}, \hat{\mathbf{S}}) \in \mathcal{A}_d}{\operatorname{argmin}} \left[\int_{\Omega} \left(\psi(\hat{\mathbf{u}}; \mathbf{p}^{(n)}) + \psi^*(\hat{\mathbf{S}}; \mathbf{p}^{(n)}) - \hat{\mathbf{S}} : \mathbf{E}(\hat{\mathbf{u}}) \right) d\Omega + \alpha' \frac{1}{n_{obs} \sigma^2} \|\Pi \mathbf{E}(\hat{\mathbf{u}}) - \mathbf{E}_{obs}\|^2 \right] \quad (2.16)$$

In order for $\hat{\mathbf{u}}$ not to fit the observations below the noise level, the order of magnitude of $\frac{1}{n_{obs} \sigma^2} \|\Pi \mathbf{E}(\hat{\mathbf{u}}) - \mathbf{E}_{obs}\|^2$ should be 1, as stated in the Morozov criterion (Morozov, 1968). Before the training of the neural network, several Steps 1 are performed to find the proper value of α' . The greater α' the lower the data discrepancy error after Step 1. This tuning is performed by dichotomy. During the minimization of the mCRE, as soon as the normalized data-driven discrepancy is out of the admissible bound after Step 1, α' is returned following this procedure and a new Step 1 is performed with the new α' .

This simple technique is powerful to prevent the network from overfitting as the predicted field \mathbf{u} cannot fit the observations below the noise level.

Remark

Here, since the value of α is fixed to enforce that $\hat{\mathbf{u}}$ is always up to noise level to the observations (before each computation of the gradient step), another choice of formulation could have been made in which α could play the role of the Lagrange multiplier. Step 1 could be modified to perform a constraint minimization to enforce that $\hat{\mathbf{u}}$ is always up to noise level of the observations. This approach would provide sufficient flexibility to be robust to noise without the need to choose the value of α by dichotomy.

3.3 Reducing the sensitivity to learning rate: automatic tuning of learning rate

Another well-known sensitive parameter in neural networks training is the choice of the learning rate l_r (Jacobs, 1988). A too small value of l_r can cause the training to be extremely slow, whereas a too large value can cause instability during the training with the risk of not reaching convergence. The value of the learning rate should be analyzed with regard to the progress made during one training step. Here, the two steps minimization procedure provides an interesting indicator on the update speed of the training. Indeed, the number of iterations performed in Step 1 depends on the modification of the space of solutions that satisfy the constitutive relation for given parameters (Γ_p) caused by the parameters update in Step 2 (because Step 1 is initialized with $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ found at the end of the previous Step 1), as it is shown in Figure 2.2.

According to multiple experiments, a good compromise is located around 4 iterations

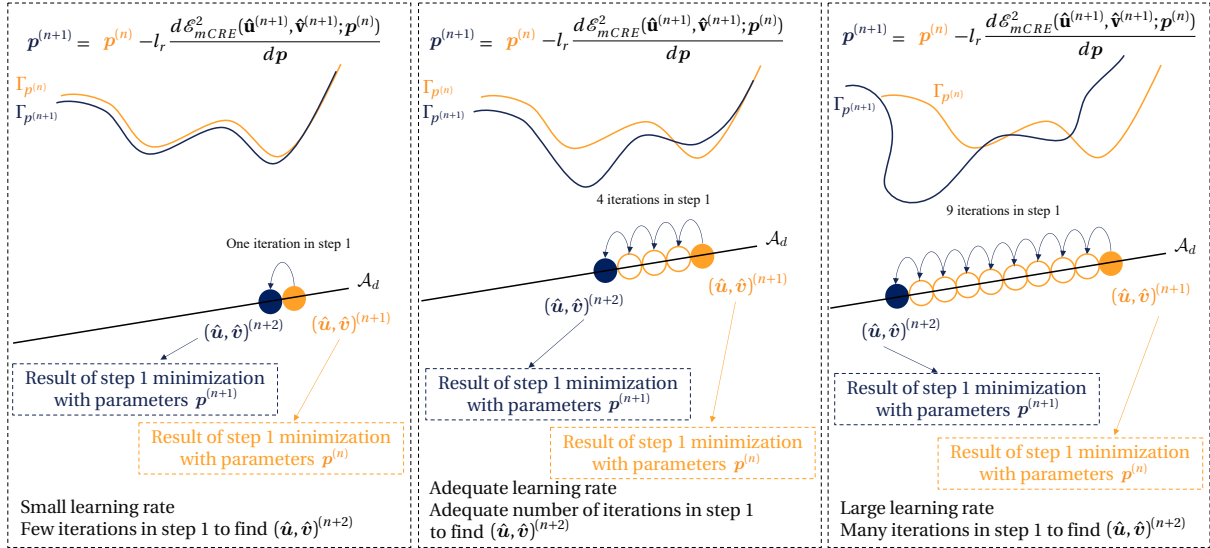


Figure 2.2 • Influence of the learning rate on the number of iterations in Step 1.

in Step 1. It is thus possible to automatically adapt the learning rate with the following empirical-based rule:

$$lr \leftarrow lr \times \text{update_coefficient}(\text{number_iterations_step_1}) \quad (2.17)$$

An example of the values of $\text{update_coefficient}(\text{number_iterations_step_1})$ is presented in Figure 2.3. These values have been obtained empirically by testing a significant amount of values on different test cases compared to the ones presented in this paper. The main advantage of this rule is that the convergence of the method is no more sensitive to the user learning rate choice, as shown in Section 5.3.1, which enables to train the network online. Note that this rule can be adapted depending on the test case: a way to adapt this rule is to set the target number of iterations to the number of iterations needed to solve the FEM problem for the initialized network.

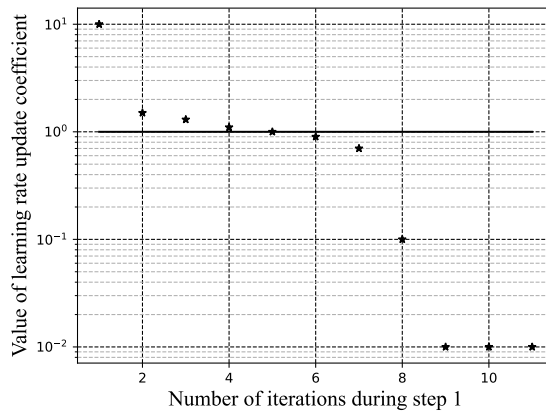


Figure 2.3 • Example of the value of the coefficient used in the empirical learning rate update rule as function of the number of iterations during Step 1.

3.4 Reducing the sensitivity to the number of epochs: definition of a physics-based stop criterion

Neural networks training requires a stop criterion. A naive technique, but yet efficient in many approaches, is to predefine a number of epochs before the training and to retune this parameter *a posteriori* as long as the predictions are not satisfactory (Goodfellow et al., 2016). This technique is not suited here as the training should be performed online for DDDAS. Another widely used rule is to define a criterion to prevent overfitting. The idea is to stop the training when the loss computed on the validation dataset is stagnating while the loss on the training dataset is still decreasing. This criterion can not work here because of the unsupervised nature of the training.

The mCRE minimization is a really interesting framework as it provides a strong physical sense that can be used to define a stop criterion. In the case of neural networks that are theoretically universal approximators, the method can be designed as if there were no model bias. Yet, to the authors' knowledge, this property has not yet been shown in the case of physics-augmented constitutive modeling. The assumption here is that a physics-augmented neural network (constructed to enforce in the architecture all the requirements of a constitutive model) can approximate any physics-consistent constitutive model (which matches the same requirements). Assuming there is no model bias, the network can represent any constitutive model that satisfies the imposed constraints. As this is further shown in Section 5.1.3, the CRE term can be interpreted as a modeling error. The CRE is homogeneous to an energy and can thus be compared to the energy in the structure, the normalized CRE writes:

$$\mathcal{E}_{\text{normalized_CRE}}^2 = \frac{\mathcal{E}_{CRE}^2}{\int_{\Omega} \psi(\mathbf{E}(\mathbf{v})) d\Omega} \quad (2.18)$$

A stop criterion (for the overall convergence of the method) can then be established based on the relative energy error in the system:

$$\mathcal{E}_{\text{normalized_CRE}}^2 < tol \quad (2.19)$$

where *tol* is a user-defined target value of the stop criterion.

This criterion is also used to estimate how far from convergence is the minimization process: near convergence the number of iterations in Step 1 is naturally lower. This can be a problem in the previous learning rate update rule, because the learning rate can be set to high values which can lead to a leap far from the minimum. To alleviate this concern, near convergence the learning rate update rule is modified so that the learning rate is only updated if it tends to decrease: the update coefficients larger than 1 are set to 1 near convergence.

3.5 Extension to the case of multiple loadings in the database

So far, the method has been presented in the case of only one loading case. In practice, the training database is constituted of multiples loading cases so the method should be modified:

- Step 1 and Step 2 are consecutively performed for each loading case.
- For the stop criterion, the method stops when the criterion is reached on all the loading cases. When a loading case reaches the criterion, the weights and biases of the neural network are no longer updated with this loading case. In each epoch, a verification is made to check if the criterion is still reached for this loading case: if the criterion is not reached, the weights and biases are then updated.
- Concerning the learning rate, the same value is applied for all the loading cases. The update rule is applied after each epoch (when all the loading cases have been treated) and takes into account the mean of iterations performed in all steps 1.

4 Practical implementation

The implementation of this method was done using the PyTorch library in Python. PyTorch is a powerful open-source Machine Learning library developed by Facebook's AI Research Lab (FAIR) that has gained widespread acceptance for its ease of use, flexibility, and computational efficiency, especially in the areas of Deep Learning and artificial intelligence.

The library is well known for its automatic differentiation system, implemented through a component called Autograd, which automates the calculation of gradients - a critical feature for training neural networks. This feature simplifies the development of complex models by abstracting the complex mathematics involved in backpropagation, making it accessible to those with a limited understanding of the underlying theory.

PyTorch supports GPU acceleration, significantly speeding up computations and facilitating the handling of large datasets, which is critical for Deep Learning tasks. This is achieved through seamless CUDA integration, enabling efficient parallel computation and batch processing capabilities.

The library's modular design gives researchers and developers the flexibility to add their own components or use the extensive set of pre-built modules to create neural networks, optimizers, and loss functions. This modularity supports a wide range of applications, from simple linear regression models to complex neural networks.

Batch operations are also an important feature of PyTorch, allowing vectorization of operations and efficient handling of large datasets by dividing them into manageable batches. This feature is critical for training models on large data sets, as it helps minimize memory usage while maximizing computational efficiency.

The coupling of the mCRE with PyTorch was quite challenging. In fact, the minimization of the mCRE requires the implementation of a finite element solver, which was implemented

from scratch using PyTorch. The main feature of this FE code is that it supports the automatic differentiation of a thermodynamic potential. This is necessary because in the most general case where a potential is described by a multilayer neural network, the derivatives of the potential are too complicated to be obtained analytically. During the 3 years of this thesis, the Pytorch library has evolved a lot. An important step is the integration of the library developed in (Horace He, 2021) into PyTorch, which now makes the implementation easier. In the implemented version of the mCRE minimization code in 2024, all operations, including the first and second derivatives of the potential and the matrix assembly, are vectorized, which provides a very important speedup (about 500 times faster than the non-vectorized version). Since each gradient descent step requires the computation of admissible fields, this efficient implementation is very important. However, for the size of the problem treated in this manuscript, the GPU implementation did not give any speed-up, because the computation of admissible fields requires a matrix inversion, which is faster on the CPU. For the problems treated in the manuscript, performing all computations on the GPU except the inversion on the CPU did not give any speedup because of the time needed to switch data from the GPU to the CPU and back to the GPU.

5 Results

5.1 Application on a nonlinear case in small deformation

This part illustrates the performance of the method on a first test case, with a non-quadratic potential to be learned under the small strain regime. Section 5.1.1 and 5.1.2 respectively present the training and validation databases, Section 5.1.3 presents the constitutive relation error before the training of the neural network with the mCRE procedure (*i.e.* after the initialization training), and Section 5.1.4 proposes an evaluation of the method on this test case.

5.1.1 Training database

The training database is constituted of synthetic data generated with a finite element simulation:

- **constitutive relation:** the chosen reference non-quadratic potential distinguishes the behavior in tension from the one in compression along the longitudinal axis (axis 1):

$$\psi(\boldsymbol{\epsilon}) = \frac{1}{2} \tilde{E} \langle \epsilon_{11} \rangle_+^2 + \frac{1}{2} E \langle \epsilon_{11} \rangle_-^2 + \frac{1}{2} E \epsilon_{22}^2 + G \epsilon_{21}^2 \quad (2.20)$$

where $\langle \bullet \rangle_+$ and $\langle \bullet \rangle_-$ stand respectively for the positive and negative parts, $\tilde{E} = 12GPa$ and $E = 20GPa$ are the tension and compression Young moduli, $G = 8GPa$ is the Coulomb modulus and ϵ_{ij} are components of the linearized strain tensor $\boldsymbol{\epsilon} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$.

- geometry: the geometry considered is the 2D (plane strain) cantilever beam presented in Figure 2.4.
- loading: the database includes 10 different loading cases \mathbf{f}_d^s with a combination of pure bending, tension and compression.

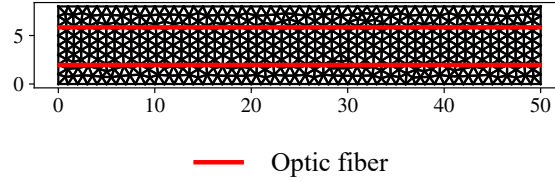


Figure 2.4 • Geometry, mesh and position of optic fibers.

After the finite element simulation, the component of the strain in the direction of the optic fiber is saved. White Gaussian noise is then added to the synthetic measurements, with a level discussed in Section 5.3.3. It is worth noticing that only the noisy observations and boundary conditions are used in the mCRE minimization procedure: the analytical expression of the target potential is not used in the training but only for validation purposes. Thus this training can be qualified as unsupervised training.

5.1.2 Validation database

The validation database is constituted of 6 deformation paths: uniaxial tension (UT), uniaxial compression (UC), biaxial tension (BT), biaxial compression (BC), simple shear (SS), and pure shear (PS). Equations (2.21) show the deformation gradient for these 6 cases, with $\gamma \in [0, 1e^{-3}]$ a loading parameter. 100 points per loading are used in the validation database. This validation process is the same as the one used in (Thakolkaran et al., 2022). These deformation paths are only used for validation and are not used during training. For validation, a comparison is made between the learned and the ground truth potential along these deformation paths.

$$\mathbf{F}^{UT}(\gamma) = \begin{bmatrix} 1+\gamma & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{F}^{UC}(\gamma) = \begin{bmatrix} \frac{1}{1+\gamma} & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{F}^{BT}(\gamma) = \begin{bmatrix} 1+\gamma & 0 \\ 0 & 1+\gamma \end{bmatrix},$$

$$\mathbf{F}^{BC}(\gamma) = \begin{bmatrix} \frac{1}{1+\gamma} & 0 \\ 0 & \frac{1}{1+\gamma} \end{bmatrix}, \mathbf{F}^{SS}(\gamma) = \begin{bmatrix} 1 & \gamma \\ 0 & 1 \end{bmatrix}, \mathbf{F}^{PS}(\gamma) = \begin{bmatrix} 1+\gamma & 0 \\ 0 & \frac{1}{1+\gamma} \end{bmatrix} \quad (2.21)$$

5.1.3 Error after initialization training

Before minimizing the mCRE, the network is initialized to represent linear isotropic elasticity thanks to a classical supervised training described in Section 3.1 The potential used to generate data for the initialization training is the following:

$$\psi(\epsilon) = \frac{1}{2}E \epsilon_{11}^2 + \frac{1}{2}E \epsilon_{22}^2 + G \epsilon_{21}^2 \quad (2.22)$$

with $E = 24GPa$ the Young modulus, $G = 11GPa$ the Coulomb modulus and ϵ_{ij} the components of the linearized strain tensor.

| | |
|--|--|
| Number of layers | 4 |
| Number of neurons per hidden layer | 50 |
| Optimizer | Adam |
| Number of epochs | 1e5 |
| Learning rate (cyclic l_r scheduler Smith, 2017) | $l_r \in [1e^{-5}, 1e^{-4}]$, cycle of 100 epochs |

Table 2.1 • Parameters used for the initialization training.

Table 2.1 summarizes the parameters used for this initialization training. At the end of this training, the network is well suited to represent linear elasticity as shown in Figure 2.5 (top). However, along axis 1, this potential does not represent well the non-quadratic potential: this is the goal of the mCRE minimization to train this neural network to accurately represent the non-quadratic potential (correction of model bias). Figure 2.6 shows, for a bending loading, the fields \mathbf{S}_{11} and ϵ_{11} associated to displacement fields $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$. To understand the role of fields $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ the reader can refer to the linear elastic case in 1.3. Figure 2.6 demonstrates that the discrepancy between fields associated with $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ are localized where the ϵ_{11} is positive, *i.e.* where the material is in tension along axis 1. This discrepancy is explained by the difference between the non-quadratic potential and the neural network in tension that can be observed in Figure 2.5. At the bottom of Figure 2.6 the normalized CRE map shows the localization of the error. This CRE map is a powerful tool because:

- the analytical ground truth potential is not needed to compute the error,
- an error level is available everywhere in the structure, even where there is no observation,
- it can deal with noisy and partial observations,
- this error is localized in the zone of model bias (here in the zone in tension along axis 1),

For all these reasons, the CRE is a strong indicator of model bias, and can even be used in the inference phase. In the following, results after the minimization of mCRE are presented and discussed.

5.1.4 Evaluation of the method

Figure 2.5 (bottom) shows the potential predicted by the neural network before and after the mCRE minimization: the model bias is corrected as the tension behavior along axis 1 is now correctly predicted. This correction of model bias is possible as the loading database

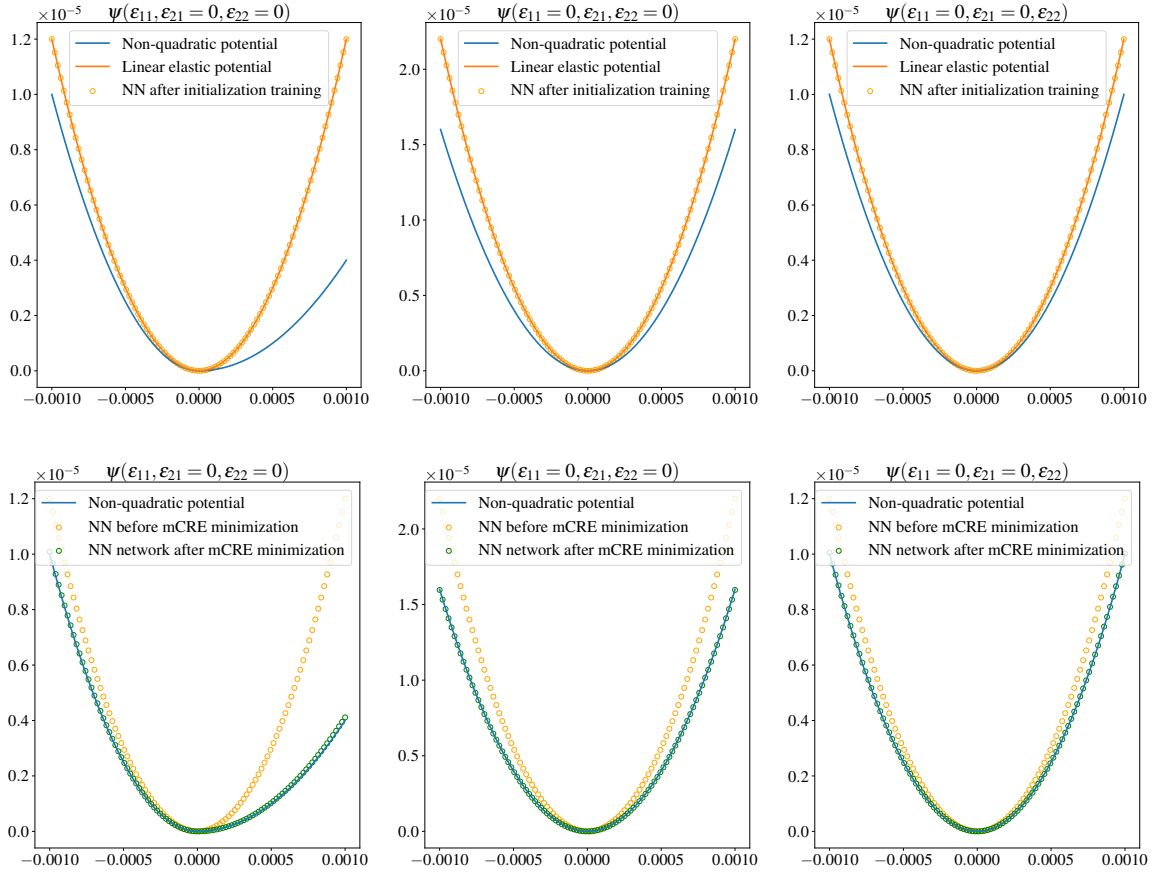


Figure 2.5 • Top: Potentials evolution with respect to the components of the linearized strain tensor after the initialization training to represent linear elasticity. This is the prediction of the neural network before the mCRE minimization. Bottom: Potentials evolution with respect to the components of the linearized strain tensor before and after mCRE minimization.

contains solicitations that make appear tension along axis 1. Figure 2.7 shows the convergence of the mean (over the loading cases) of the mCRE and normalized CRE as well as the evolution of relative error on the validation dataset. The similarity in the evolution between mCRE and normalized CRE is explained by the fact that α is adapted during training to constraint the normalized data discrepancy term of the mCRE to be close to 1 (see Section 3.2).

About the evolution of these criteria, the plateau at the beginning of training is caused by the small value of the initial learning rate. With the adaptive learning rate rule described in Section 3.3, the learning rate gets high enough to achieve decreasing of the mCRE after several epochs (here around 12 epochs).

5.2 Application on a hyperelastic case: Mooney-Rivlin model

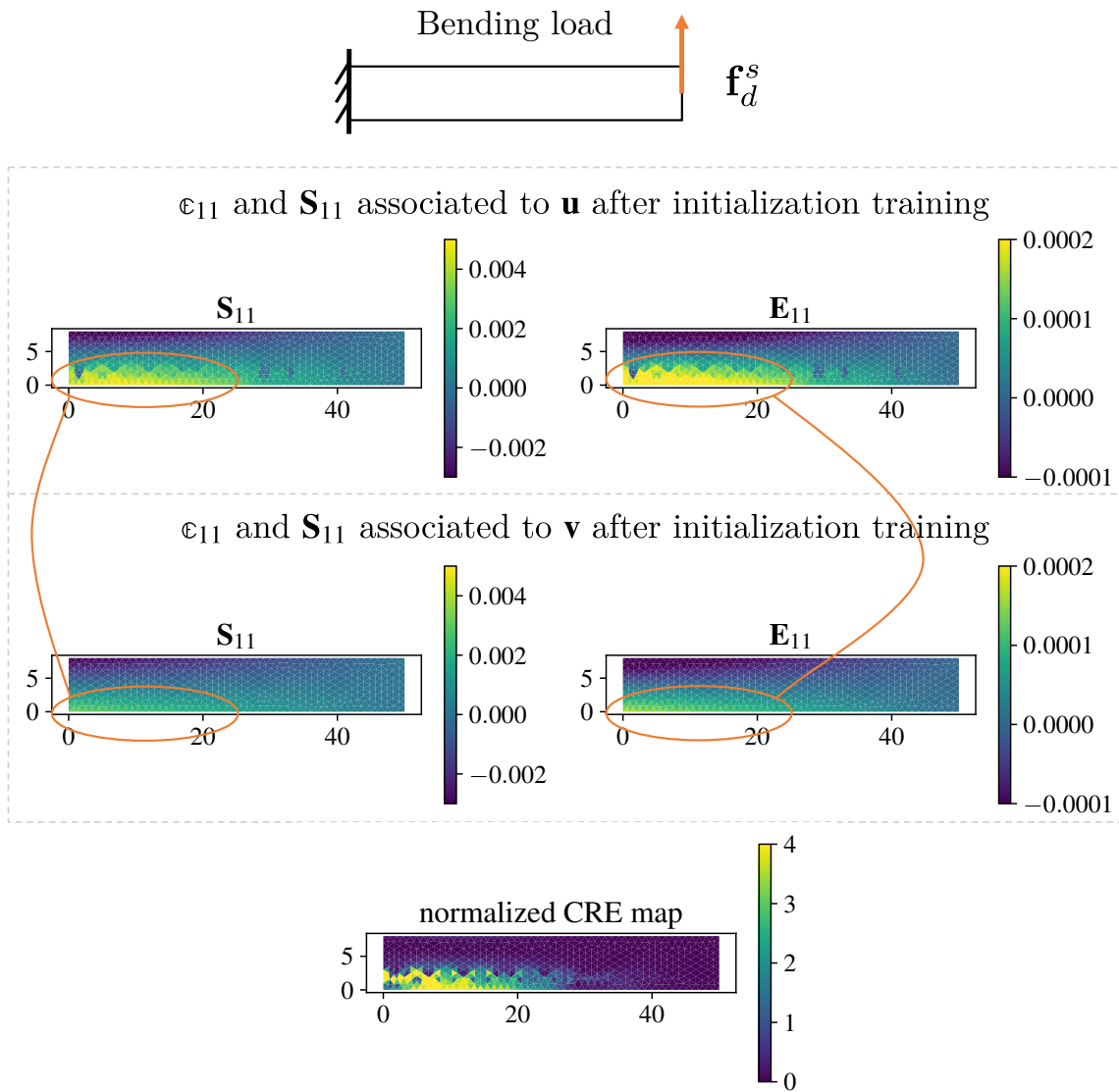


Figure 2.6 • For a bending loading (top), fields \mathbf{S}_{11} and ϵ_{11} associated with displacement fields $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ (middle) and the normalized CRE map (bottom). These observations depend on the initialization choice.

5.2.1 Training database

The training database is constituted of synthetic data generated with a finite element simulation:

- constitutive relation: Mooney-Rivlin potential (Mooney, 1940; Rivlin, 1948):

$$\psi(\mathbf{C}) = A(\tilde{\mathbf{I}}_1 - 3) + B(\tilde{\mathbf{I}}_2 - 3) + C(\tilde{\mathbf{I}}_1 - 3)(\tilde{\mathbf{I}}_2 - 3) \quad (2.23)$$

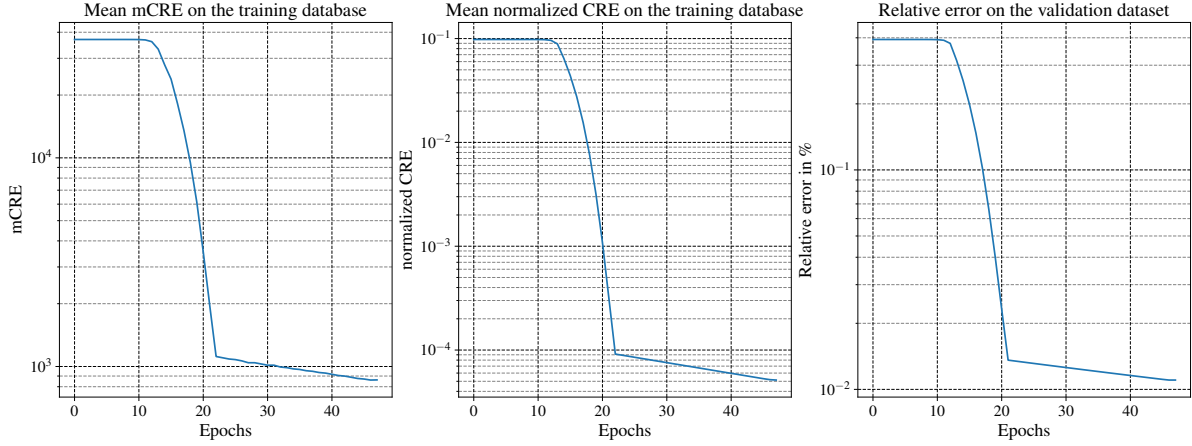


Figure 2.7 • left: evolution of the mean (over the loading cases) mCRE during the training; center: evolution of the mean (over the loading cases) normalized CRE during the training, right: evolution of the relative error on the validation database during training.

with

$$\begin{aligned}
 J &= \det(\mathbf{C}) \\
 \tilde{I}_1 &= J^{-2/3} \text{tr}(\mathbf{C}) \\
 \tilde{I}_2 &= \frac{1}{2} J^{-4/3} [\text{tr}(\mathbf{C})^2 - \text{tr}(\mathbf{C}^2)] \\
 \mathbf{C} &= 2\mathbf{E} + \mathbf{I}
 \end{aligned}$$

and $A = 1.9$, $B = 0.4$ and $C = 1$ some material parameters.

- geometry: the geometry considered is a 2D (plane strain) square.
- loading: the database includes 30 different loading cases \mathbf{f}_d^s with uniaxial tension and uniaxial compression. The training database is deliberately chosen with few loading cases to quantify the generalization to out-of-training database loading cases.

After the finite element simulation, the full displacement is stored and white Gaussian noise is then added to the synthetic measurements with a noise level of 0, 1%. Here again, only the noisy observations and boundary conditions are used in the mCRE minimization procedure: the analytical expression of the target potential is not used in the training but only for validation purposes. Even though the method presented in Section 2 is detailed in the case of strain observations, it extends easily to the case of displacement observations.

5.2.2 Validation database

The validation database is constructed according to the same procedure as the one described in Section 5.1.2 with the deformation paths (2.21). The difference is that here, the target model describes hyperelastic behavior so the range of γ should be different: $\gamma \in [0, 1]$.

5.2.3 Evaluation of the method, case of incomplete training database

For this test case in which the network has to learn a Mooney-Rivlin constitutive model, the method is evaluated when the training database does not include enough deformation paths to describe all features of the constitutive model. The interest of this incomplete training database is to see the generalization outside the training database. Before the minimization of the mCRE, the network is initialized thanks to an initialization training described in Section 3.1 with the Neo-Hookean model:

$$\psi(\mathbf{E}) = A(\tilde{I}_1 - 3) + B(J - 1)^2 \quad (2.24)$$

with $A = 1.9$ and $B = 2.4$ are material parameters.

Figure 2.8 shows the potential before and after the mCRE minimization along two deformation paths. On the left, it is uniaxial compression along one axis (whereas it was the other axis in the training database) and the learned potential is close to the target. This generalization from one axis to another is explained by the fact that the isotropic invariants were put inside the network. On the right, the response to pure shear is not correctly described as this does not appear in the training database. Yet, this lack of precision is compensated by the mCRE framework that can be used to assimilate data in the online phase by the mean of the predicted fields associated with \mathbf{u} (trade-off between the prediction from the trained model and assimilated data). Moreover, even in the prediction phase when data are assimilated the CRE modeling error is available to give a confidence bound on the prediction.

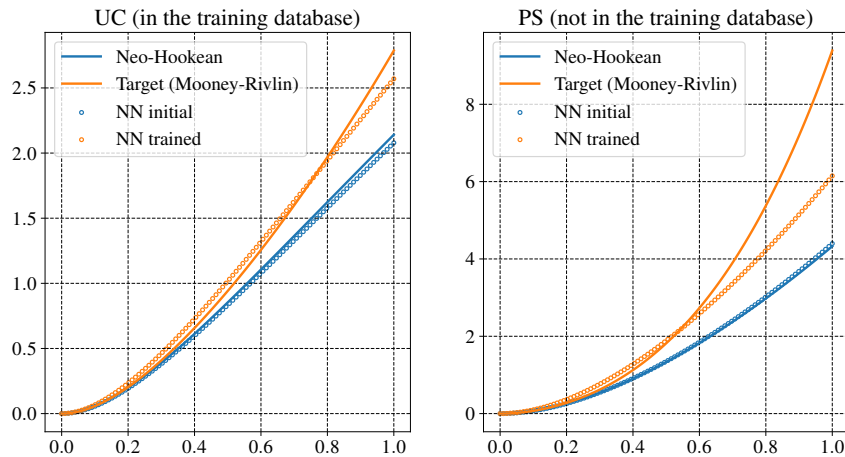


Figure 2.8 • Potential before and after the mCRE minimization along two deformation paths: on the left is uniaxial compression and on the right is pure shear.

5.3 Robustness on hyperparameters choice

5.3.1 Relevance of adaptive learning rate

The learning rate parameter is a crucial parameter for the convergence of Deep Learning training: Figure 2.9 (left) shows the evolution of the relative error on the validation dataset during the training for several different values of l_r . This figure highlights the sensitivity of this parameter on good convergence. In most Deep Learning use cases, this parameter is tuned by the user until the results are satisfactory. Here, the DDDAS context (Chamoin, 2021) imposes that the training has to be performed online, thus imposing an automatic tuning of this parameter. Figure 2.9 (middle) shows the evolution of the relative error on the validation dataset for different initial learning rates. This shows that the adaptive learning rate rule drastically reduces the sensitivity of the choice of the learning rate as very different initial learning rates achieve close convergence whereas it was not the case without the adaptive learning rate. Figure 2.9 (right) demonstrates that the learning rates all stabilize to a close value, even with an important difference in the initial learning rate choice. This is the case only when the initial learning rate is small: if the initial learning rate is too large, the first epoch will update the network far from the initialization with *a priori* knowledge, so it is better to set a low initial value. Indeed, even if there is a high difference between the low initial learning rate and the proper one, the learning rate will adapt only in a few iterations.

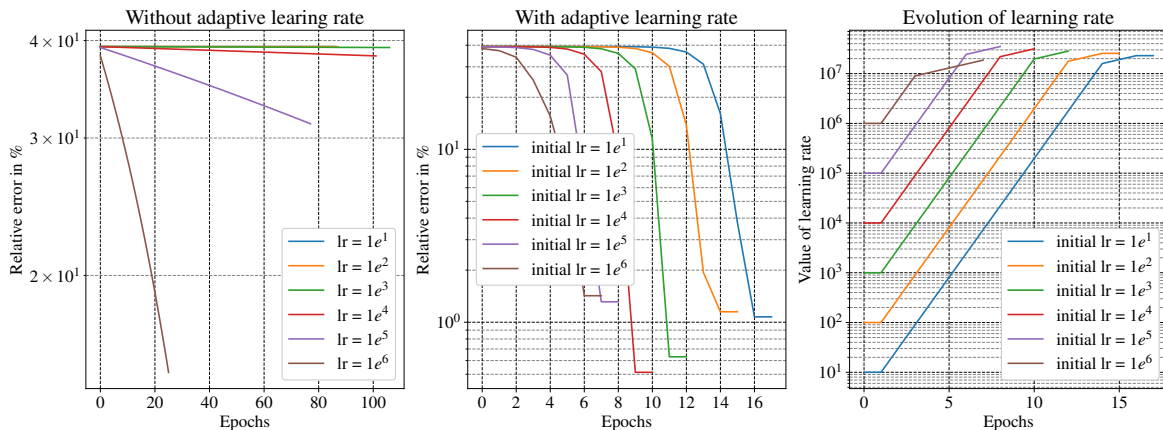


Figure 2.9 • Left and middle: evolution of the relative error on the validation dataset (left: sensitivity of the fixed learning rate without adaptive learning rate, middle: sensitivity of the initial learning rate when the adaptive rule is applied). Right: evolution of the learning rate when the adaptive rule is applied.

5.3.2 Relevance of adaptive weighting between losses

This part presents the interest of the adaptive tuning of α . First, Figure 2.10 helps to understand the role played by α . It shows the strain and stress fields associated with displacement fields $\hat{\mathbf{u}}$ at the end of Step 1 for different values of α . Equation (1.13) shows the role of α in the linear elastic case: $\hat{\mathbf{u}}$ is a compromise between the observed data and the model for the current parameters, and α sets up the balance between these two terms. A similar interpretation can be done in the case of non-quadratic potentials as shown in Figure 2.10:

- For a small value of α (here when $\alpha = 1$), \mathbf{E}_{11} and \mathbf{S}_{11} are really close to the one obtained with $\hat{\mathbf{v}}$ (no compromise with data). The associated CRE map shows there is no error, which means the value of α is too low.
- On the opposite, $\alpha = 10\,000$ shows an important difference between \mathbf{u} and \mathbf{v} but the solution associated with $\hat{\mathbf{u}}$ seems noisy.
- In this case a good choice of α is between 100 and 10 000: the automatic tuning gives a value of $\alpha = 2\,000$.

Figure 2.11 (left) illustrates how the convergence is affected by different values of α . This shows that convergence is sensitive to the chosen values of α , which motivates the need for adaptive tuning. The right-hand side of Figure 2.11 shows the convergence for the same initial values of α . As α is always chosen such that the $\hat{\mathbf{u}}$ field fits the observations to the noise level, the convergence is the same for these 3 initial values of α .

Nevertheless, this adaptive tuning comes with a high computational cost: when $\hat{\mathbf{u}}$ does not fit the observations to noise level, multiple Steps 1 have to be performed in the dichotomy to find the proper value of α . This constitutes an important drawback of the method and should be further investigated.

5.3.3 Robustness to noise

The mCRE for parameter identification has already been shown to be very effective in the presence of noisy measurement (Allix et al., 2005; Feissel & Allix, 2007). This article confirms the previous observations in the case of neural network training. Figure 2.12 (left) shows the evolution relative error on the validation dataset for different noise levels: even with high noise levels, the relative error on the validation dataset is acceptable. Figure 2.12 (right) shows the potential learned with high noise. For the 40% noise case, the potential is not representative of reality but is still close and better than before the training. This is achievable thanks to the adaptive α tuning which requires the knowledge of the noise level.

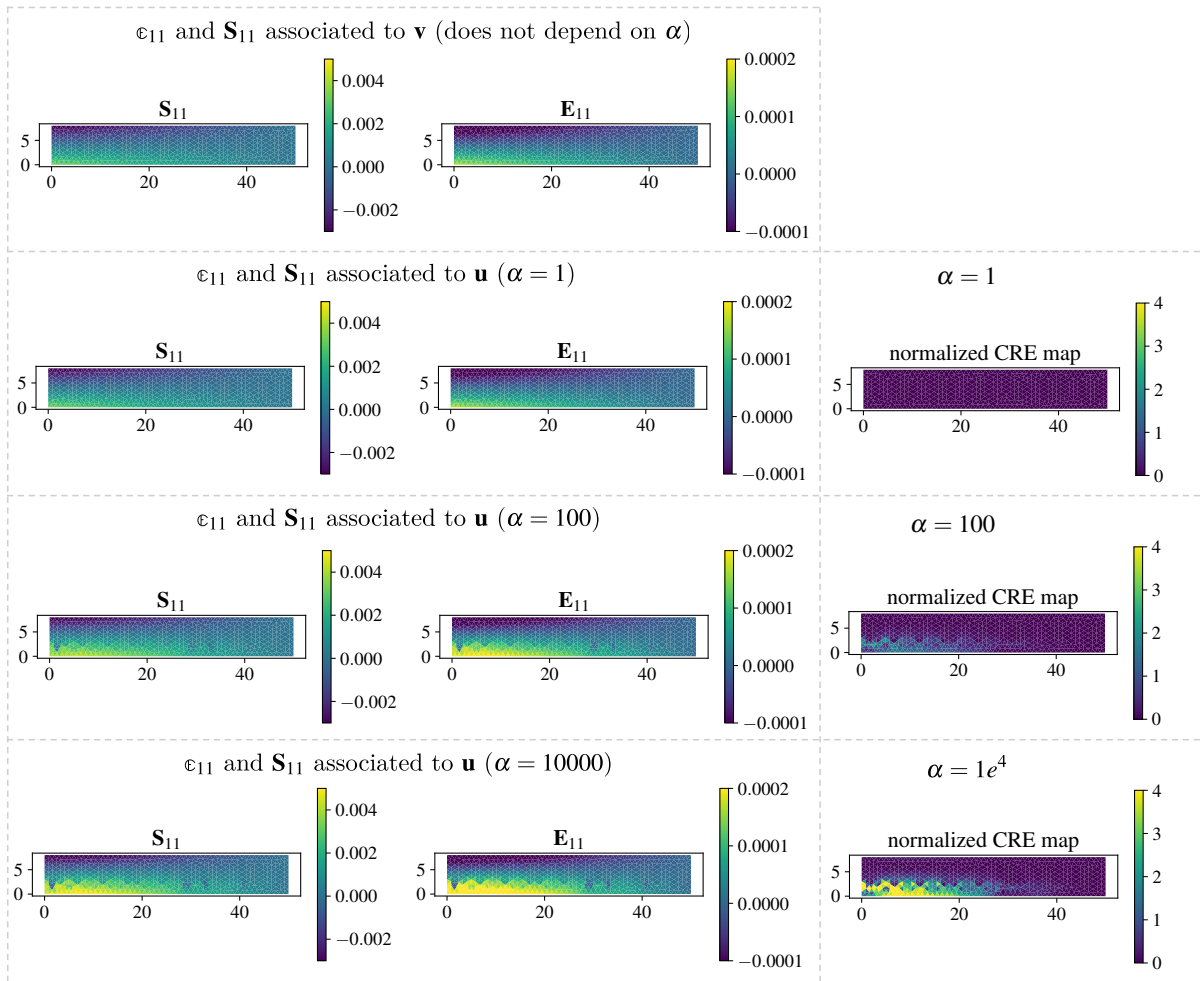


Figure 2.10 • Influence of α on the strain and stress fields associated to displacement field $\hat{\mathbf{u}}$ at the end of Step 1 (see Section 1.1.1) and on the associated CRE map.

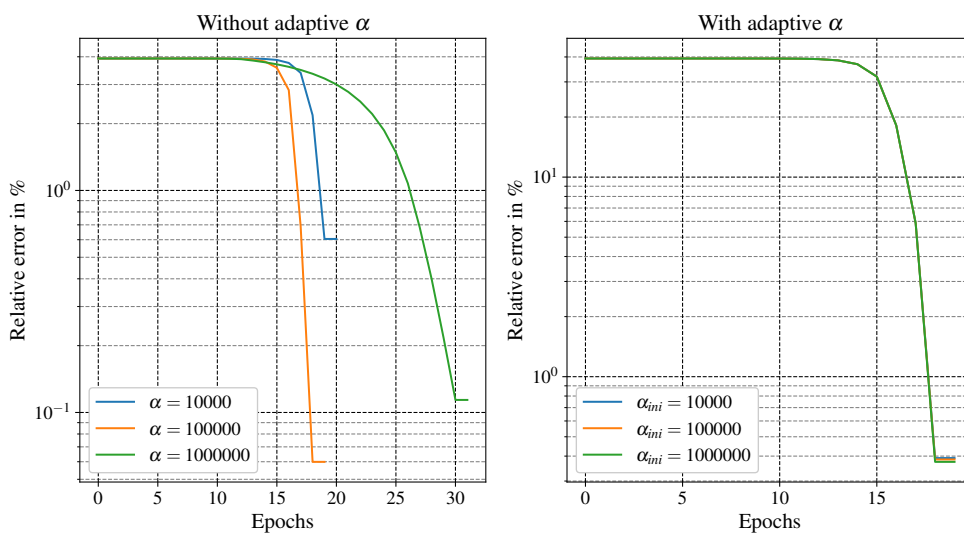


Figure 2.11 • Evolution of the relative error on the validation dataset. Left: sensitivity of the parameters α without adaptive rule. Right: sensitivity of the initial α rate when the adaptive rule is applied.

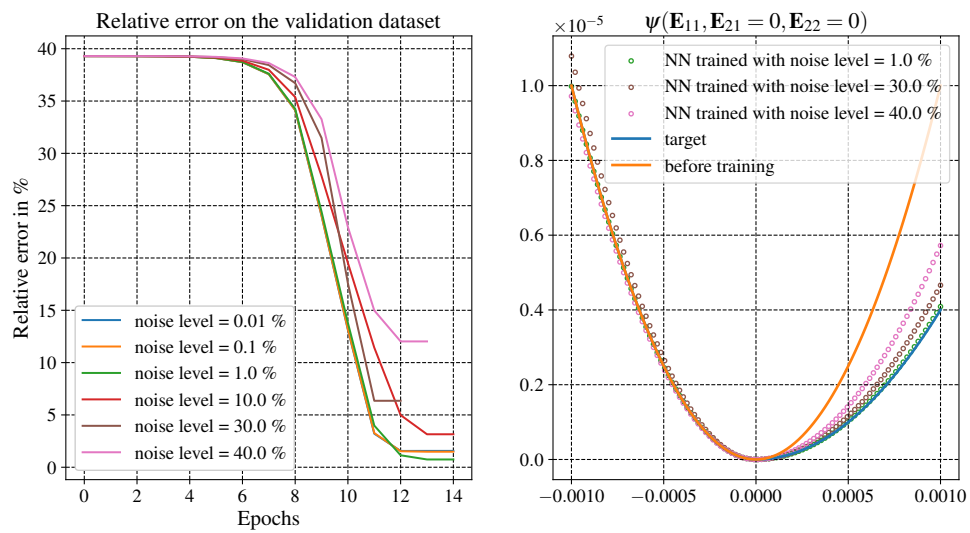


Figure 2.12 • Influence of the noise level on the quality of the training. Left: evolution of the relative error on the validation set, right: $\psi(\mathbf{E}_{11}, \mathbf{E}_{21} = 0, \mathbf{E}_{22} = 0)$ for different noise levels.

Chapter conclusion

This chapter introduced a novel approach to train physics-augmented neural networks for constitutive law representation using partial strain or displacement measurements and boundary conditions. All reliable information such as thermodynamics or equilibrium properties was enforced either in the network or in the minimization procedure. This chapter has shown that it is possible to train a neural network in an unsupervised way thanks to the modified Constitutive Relation Error (mCRE).

The automatic hyperparameter tuning is another significant feature of the proposed method, which is particularly relevant for online training in the Dynamic Data Driven Application Systems (DDDAS) paradigm. The evaluation of the method has shown to be effective in producing accurate models in the case of nonlinear state laws, excellent noise robustness and low sensitivity to user-defined hyperparameters. Overall, this proposed method offers a potentially valuable tool for predicting the behavior of materials and structures under external loadings, thus opening up new avenues for research in the field of computational mechanics. Nevertheless, at this point only a history-independent behavior has been learned. The following chapter addresses history-dependent dissipative behaviors by incorporating evolution laws in the learning process.

3

Learning nonlinear history-dependent behaviors with neural networks trained in the mCRE framework

The work presented in this chapter has been the subject of the following publication: Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Unsupervised learning of history-dependent constitutive material laws with thermodynamically-consistent neural networks in the modified Constitutive Relation Error framework. *Computer Methods in Applied Mechanics and Engineering*, 2024, 425, pp.116967. <https://dx.doi.org/10.1016/j.cma.2024.116967>

Contents

| | | |
|----------|---|-----------|
| 1 | Problem definition | 64 |
| 1.1 | A minimization procedure suited for history-dependent behavior . | 65 |
| 1.1.1 | First step: computing the optimal admissible solution for given parameters | 66 |
| 1.1.2 | Second step: update of parameters | 68 |
| 2 | Training NN with the mCRE framework | 69 |
| 2.1 | Constitutive model described by a thermodynamically-consistent neural network. | 70 |
| 2.2 | Hyperparameters automatic tuning strategy | 70 |
| 2.2.1 | Morozov-based criterion for automatic tuning of weighting between losses | 71 |
| 2.2.2 | Adaptive tuning of the learning rate | 71 |
| 3 | Results | 72 |
| 3.1 | A simple 1D example to identify an isotropic hardening modulus . . | 73 |

| | | |
|-------|--|----|
| 3.2 | Rate-independent test case: learning a nonlinear isotropic hardening law | 76 |
| 3.2.1 | Reference problem | 76 |
| 3.2.2 | mCRE minimization for the rate-independent test case. . . | 78 |
| 3.2.3 | General results | 79 |
| 3.2.4 | Robustness to noise level | 80 |
| 3.2.5 | Relevance of the automatic learning rate tuning | 82 |
| 3.3 | Rate-dependent test case: learning a dissipation pseudo-potential . | 83 |
| 3.3.1 | mCRE minimization for the rate-dependent test case | 85 |
| 3.3.2 | General results | 86 |
| 3.3.3 | Localization of the model bias in the structure | 87 |
| 3.3.4 | Relevance of the automatic tuning of the weighting between losses | 87 |

Chapter 1 introduced the mCRE formulation for material behavior within the framework of generalized standard materials. Few mCRE studies have focused on the recalibration of history-dependent model parameters, and all assumed a specific model form (Marchand et al., 2019; Bonnet et al., 2024). Following the previous chapter, this chapter aims to free the process from the traditional model bias present during parameter identification within the mCRE framework. The previous chapter was dedicated to learning nonlinear constitutive laws but was restricted to history-independent behaviors (state equations). This chapter then focuses on the training of neural networks, constrained to respect the framework of generalized standard materials, to learn history-dependent behaviors with neural networks. The method is evaluated on different test cases by means of several criteria such as the accuracy of the learned model, the evolution of the loss function during the training, the noise robustness, the localization of modeling error and the relevance of automatic hyperparameters tuning rules.

1 Problem definition

To define the problem notations, let us consider a body in an initial configuration $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$) with boundary $\partial\Omega$ and isothermal environment, observed for a period $[0, T]$ under the small strain assumptions. Dirichlet boundary conditions are imposed on $\partial\Omega_1 \subset \partial\Omega$ by means of a time-dependent displacement field \mathbf{u}_d . Neumann boundary conditions are prescribed on $\partial\Omega_2 \subset \partial\Omega$ by means of a time-dependent traction field \mathbf{f}_s^d . A time-dependent body force field \mathbf{f}_v^d may also be prescribed in Ω . Additionally, noisy strain measurements ϵ_{obs} (in the case of observations from optic fibers) or displacement measurements \mathbf{u}_{obs} (in the case of digital image correlation) are available.

The solution to the direct mechanical problem is the set of the variables $(\mathbf{e}_e, \mathbf{e}_p, \mathbf{s})$ satisfying the three following groups of equations for each time $t \in [0, T]$:

- **kinematic admissibility** defines the space \mathcal{U}_{ad} of displacement fields \mathbf{u} satisfying the Dirichlet boundary conditions:

$$\mathbf{u}|_{\partial\Omega_1} = \mathbf{u}_d \quad (3.1)$$

- **static admissibility** defines the space \mathcal{S}_{ad} of stress fields satisfying the equilibrium:

$$\int_{\Omega} \mathbf{s} : \mathbf{e}(\mathbf{v}) d\Omega = \int_{\Omega} \mathbf{f}_d^v \cdot \mathbf{v} d\Omega + \int_{\partial\Omega_2} \mathbf{f}_d^s \cdot \mathbf{v} dS \quad \forall \mathbf{v} \in \mathcal{U}_0 \quad (3.2)$$

with $\mathbf{e} = \mathbf{e}_e + \mathbf{e}_p$, and \mathcal{U}_0 the space of kinematic admissibility with homogeneous Dirichlet conditions.

- **constitutive behavior:**

- a set of state equations:

$$\mathbf{s} = \frac{\partial \psi}{\partial \mathbf{e}_e} \quad (3.3)$$

- a set of evolution laws:

$$\mathbf{s} = \frac{\partial \varphi}{\partial \dot{\mathbf{e}}_p} \quad (3.4)$$

The potentials ψ and φ are parametrized with some parameters \mathbf{p} .

The minimization of the mCRE aims to identify the parameters \mathbf{p} of constitutive relations that fit the best to experimental data. Here Dirichlet and Neumann boundary conditions are assumed to be known (even though the framework extends to unreliable boundary conditions (Nguyen, 2021)), whereas the observations $\boldsymbol{\epsilon}_{obs}$ or \mathbf{u}_{obs} are affected by measurement noise. Here, we consider observations \mathbf{u}_{obs} .

1.1 A minimization procedure suited for history-dependent behavior

The inverse problem consists of finding the optimal parameters \mathbf{p}_{opt} (involved in the thermodynamic potentials) such that:

$$\mathbf{p}_{opt} = \underset{\mathbf{p}}{\operatorname{argmin}} \left[\min_{\hat{\mathbf{s}} \in \mathcal{A}_d} \mathcal{E}_{mCRE}^2(\hat{\mathbf{s}}; \mathbf{p}) \right] \quad (3.5)$$

with

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{s}}; \mathbf{p}) = \mathcal{E}_{CRE}^2(\hat{\mathbf{s}}; \mathbf{p}) + \frac{\alpha}{2} \int_0^T \|\Pi \hat{\mathbf{u}} - \mathbf{u}_{obs}\|^2 dt \quad (3.6)$$

where $\mathcal{A}_d = (\mathcal{U}_{ad} \times \mathcal{S}_{ad})$ (even though it is possible to consider loading as uncertain information, see (Nguyen, 2021)), α is a scaling factor, and Π is a projector of $\hat{\mathbf{u}}$ on the

sensing quantities, and:

$$\mathcal{E}_{CRE}^2(\hat{\mathbf{s}}; \mathbf{p}) = \int_0^T \int_{\Omega} \eta_{\psi}(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) d\Omega dt + \int_0^T \int_0^t \int_{\Omega} \eta_{\varphi}(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \dot{\hat{\mathbf{X}}}, \hat{\mathbf{Y}}) d\Omega ds dt \quad (3.7)$$

The minimization of the mCRE is still performed with an iterative process in which, at iteration $n + 1$:

- in Step 1, an optimal admissible solution $\hat{\mathbf{s}}^{(n+1)}$ is computed for the current parameters $\mathbf{p}^{(n)}$ such that:

$$\hat{\mathbf{s}}^{(n+1)} = \underset{\hat{\mathbf{s}} \in \mathcal{A}_d}{\operatorname{argmin}} \left[\mathcal{E}_{mCRE}^2(\hat{\mathbf{s}}; \mathbf{p}^{(n)}) \right] \quad (3.8)$$

- in Step 2, the parameters of the constitutive model are updated following a gradient descent step to get $\mathbf{p}^{(n+1)}$.

The main difference with the previous chapter is how the admissible solution is computed, because it requires the integration of evolution laws. Sections 1.1.1 and 1.1.2 respectively detail the first and second steps of this iterative minimization.

1.1.1 First step: computing the optimal admissible solution for given parameters

The first step, which is the most expensive one regarding computation time, is performed with a strategy similar to the one used in (Marchand et al., 2019; Nguyen, 2021). This strategy is inspired by the LATIN method (Ladevèze, 1999), which is non-incremental (*i.e.* global in time) and is well-suited to the mathematical structure of the mCRE. The choice made here is to split the mCRE into two positive parts $\mathcal{E}_{\psi}^2(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}})$ and $\mathcal{E}_{\varphi}^2(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \dot{\hat{\mathbf{X}}}, \hat{\mathbf{Y}})$ defined by:

$$\mathcal{E}_{\psi}^2(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) = \int_0^T \int_{\Omega} \eta_{\psi}(\hat{\boldsymbol{\epsilon}}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) d\Omega dt + \int_0^T \frac{\alpha}{2} \|\Pi \hat{\mathbf{u}} - \mathbf{u}_{obs}\|^2 dt \quad (3.9)$$

$$\mathcal{E}_{\varphi}^2(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \dot{\hat{\mathbf{X}}}, \hat{\mathbf{Y}}) = \int_0^T \int_0^t \int_{\Omega} \eta_{\varphi}(\dot{\hat{\boldsymbol{\epsilon}}}_p, \hat{\boldsymbol{\sigma}}, \dot{\hat{\mathbf{X}}}, \hat{\mathbf{Y}}) d\Omega ds dt \quad (3.10)$$

This separation into two positive parts enables minimizing each term alternatively. Equation (3.9) is a compromise between the residual on the state equations and the discrepancy with measurements. Its minimization is a linear (because elasticity is assumed to be linear here) and global in space problem. The minimization of (3.10) is local in space and corresponds to the integration of evolution laws, performed here with an Euler scheme. The general philosophy of Step 1 minimization is illustrated in Figure 3.1. \mathcal{S}_{ψ} (resp. \mathcal{S}_{φ}) is the space of variables searched in the global (resp. local) step that minimizes the quantity \mathcal{E}_{ψ} (resp. \mathcal{E}_{φ}).

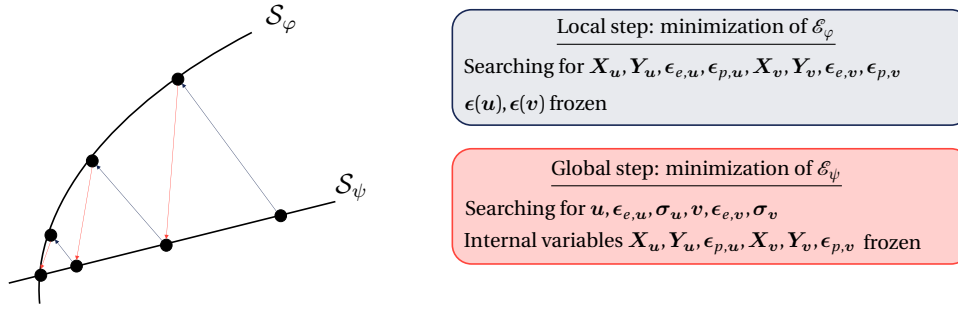


Figure 3.1 • Illustration of the LATIN-inspired scheme for Step 1 minimization of the mCRE.

Global step (minimization of \mathcal{E}_ψ^2)

In this minimization step, \mathcal{E}_ψ^2 is minimized under the admissibility constraint $(\mathbf{u}, \boldsymbol{\sigma}) \in (\mathcal{U}_{ad} \times \mathcal{S}_{ad})$. The kinematic admissibility is enforced in the search space, in which the discretization of \mathbf{u} is split into the imposed and free degrees of freedom. The static admissibility of $\boldsymbol{\sigma}$ is imposed through a Lagrangian:

$$\mathcal{L}(\hat{\epsilon}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}, \boldsymbol{\lambda}) = \mathcal{E}_\psi^2(\hat{\epsilon}_e, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}) - \int_0^T \left[\int_\Omega \hat{\boldsymbol{\sigma}} : \boldsymbol{\epsilon}(\boldsymbol{\lambda}) d\Omega - \int_\Omega \mathbf{f}_v^d \cdot \boldsymbol{\lambda} d\Omega - \int_{\partial\Omega_2} \mathbf{f}_s^d \cdot \boldsymbol{\lambda} dS \right] dt \quad (3.11)$$

In the expression of \mathcal{E}_ψ^2 , the quantity η_ψ defined in (1.33) involves a dual Legendre-Fenchel transform, defined in (1.29), which is not convenient in practice. Therefore, a displacement field $\hat{\mathbf{v}}$ (defined up to a rigid body motion) is introduced by duality such that:

$$\boldsymbol{\sigma} \in \mathcal{S}_{ad}, \quad \hat{\boldsymbol{\sigma}} = \left. \frac{\partial \psi}{\partial \boldsymbol{\epsilon}_e} \right|_{\hat{\epsilon}_{e,v}} \quad (3.12)$$

For a given $\hat{\boldsymbol{\sigma}}$, $\hat{\epsilon}_{e,v}$ is the strain providing for the supremum in the definition of the Legendre-Fenchel transform. The variables associated with the displacement $\hat{\mathbf{v}}$ (resp. $\hat{\mathbf{u}}$) are denoted $\hat{s}_v = (\hat{\epsilon}_{e,v}, \hat{\epsilon}_{p,v}, \hat{\boldsymbol{\sigma}}_v, \hat{\mathbf{X}}_v, \hat{\mathbf{Y}}_v)$ (resp. $\hat{s}_u = (\hat{\epsilon}_{e,u}, \hat{\epsilon}_{p,u}, \hat{\boldsymbol{\sigma}}_u, \hat{\mathbf{X}}_u, \hat{\mathbf{Y}}_u)$).

Replacing the Legendre-Fenchel expression involved in (1.33), (3.11) leads to:

$$\begin{aligned} \mathcal{L}(\hat{\epsilon}_{e,u}, \hat{\epsilon}_{e,v}, \hat{\boldsymbol{\sigma}}_v, \hat{\mathbf{X}}_u, \hat{\mathbf{X}}_v, \hat{\mathbf{Y}}_v, \boldsymbol{\lambda}) &= \int_0^T \int_\Omega \left[\psi(\hat{\epsilon}_{e,u}, \hat{\mathbf{X}}_u) - \psi(\hat{\epsilon}_{e,v}, \hat{\mathbf{X}}_v) - \hat{\boldsymbol{\sigma}}_v : (\hat{\epsilon}_{e,v} - \hat{\epsilon}_{e,u}) - \hat{\mathbf{Y}}_v (\hat{\mathbf{X}}_v - \hat{\mathbf{X}}_u) \right] d\Omega dt \\ &+ \int_0^T \frac{\alpha}{2} \|\Pi \mathbf{u} - \mathbf{u}_{obs}\|^2 dt \end{aligned} \quad (3.13)$$

During the global step, internal variables $\hat{\mathbf{X}}_u, \hat{\mathbf{Y}}_u, \hat{\epsilon}_{p,u}, \hat{\mathbf{X}}_v, \hat{\mathbf{Y}}_v, \hat{\epsilon}_{p,v}$ are frozen to the value obtained at the last local step. The stationarity of the Lagrangian is found with a Newton scheme similar to Chapter 2. In the case of linear elasticity, this Newton scheme converges in one iteration.

Local step (minimization of \mathcal{E}_φ^2)

In this step, the minimization of \mathcal{E}_φ^2 corresponds to the integration of evolution laws at each Gauss point (local step). This step can be parallelized. The integration of the evolution laws is performed with an Euler scheme with the total strains $\epsilon(\mathbf{u})$ and $\epsilon(\mathbf{v})$ frozen. The initial conditions on internal variables are enforced for the first time step. As mentioned in Section 1.5.1 dealing with the thermodynamic framework, in the rate-independent case the dissipation potential is the indicator of a convex domain and is not differentiable, whereas in the rate-dependent case, this potential is differentiable. This leads to different formulations in the integration of evolution laws. The local steps are detailed for each case in Section 3.

Stop criterion for Step 1

The stop criterion used for Step 1 (alternation of local and global steps) needs to be defined. This minimization is a fixed point algorithm, so the stopping criterion is defined regarding the stagnation of the CRE between two successive local and global steps. The tolerance is defined by the user and its influence will be discussed in the section dedicated to the automatic tuning of the learning rate. The last step performed needs to be the global step so that the solution of the minimization is statically admissible.

Restart strategy

After an update of the model parameters, the new Step 1 is initialized with the solutions s_u , s_v obtained at the previous Step 1. This restart strategy significantly reduces the number of iterations required for convergence, thus decreasing computation time.

1.1.2 Second step: update of parameters

This step consists of the updating of parameters \mathbf{p} with a gradient descent step:

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} - l_r \frac{\partial \mathcal{E}_{mCRE}^2(\hat{\mathbf{s}}^{(n+1)}; \mathbf{p}^{(n)})}{\partial \mathbf{p}} \quad (3.14)$$

with $\hat{\mathbf{s}}^{(n+1)}$ the solution obtained at the end of Step 1 of the iteration $n + 1$ of the mCRE minimization.

To summarize this section and introduce the next one, Figure 3.2 illustrates the general methodology for further training the physics-augmented neural network with the mCRE. This minimization can be seen as a Sequential Quadratic Programming (SQP) approach (Nocedal & Wright, 2006).

Minimization of mCRE:

$$\mathbf{p}_{opt} = \underset{\mathbf{p}}{\operatorname{argmin}} \left[\min_{\hat{s} \in \mathcal{A}_d} \mathcal{E}_{mCRE}^2(\hat{s}; \mathbf{p}) \right]$$

While $\mathcal{E}_{CRE} > \epsilon_{stop}$

Step 1:
$$\hat{s}^{(n+1)} = \underset{\hat{s} \in \mathcal{A}_d}{\operatorname{argmin}} \left[\mathcal{E}_{mCRE}^2(\hat{s}; \mathbf{p}^{(n)}) \right]$$

Elastic initialization for the first iteration or restart with the solution from last iteration

While stopping criterion is not reached :

Local step: Integration of evolution laws. Minimization of

$$\mathcal{E}_{\varphi}^2(\dot{\epsilon}_p, \hat{\sigma}, \dot{\mathbf{X}}, \dot{\mathbf{Y}}) = \int_0^T \int_0^t \int_{\Omega} \varphi(\dot{\epsilon}_p, -\dot{\mathbf{X}}) + \varphi^*(\hat{\sigma}, \dot{\mathbf{Y}}) - \hat{\sigma} : \dot{\epsilon}_p + \dot{\mathbf{X}} \cdot \dot{\mathbf{Y}} d\Omega ds dt$$

Global step: Global equilibrium

$$\mathcal{E}_{\psi}^2(\epsilon_e, \hat{\sigma}, \dot{\mathbf{X}}, \dot{\mathbf{Y}}) = \int_0^T \int_{\Omega} \psi(\epsilon_e, \dot{\mathbf{X}}) + \psi^*(\hat{\sigma}, \dot{\mathbf{Y}}) - \hat{\sigma} : \epsilon_e - \dot{\mathbf{X}} \cdot \dot{\mathbf{Y}} d\Omega dt + \frac{\alpha}{2} \int_0^T \|\Pi \hat{u} - u_{obs}\|^2 dt$$

- Kinematic admissibility imposed by the searched space
- Static admissibility imposed by Lagrangian

Step 2:

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} - l_r \frac{\partial \mathcal{E}_{mCRE}^2(\hat{s}^{(n+1)}; \mathbf{p}^{(n)})}{\partial \mathbf{p}}$$

- Gradient computed with adjoint state method

Figure 3.2 • Description of the method developed.

2 Training of thermodynamically-consistent neural networks with the mCRE framework

The previous section has recalled basics on the minimization of the modified Constitutive Relation Error in the case of a nonlinear constitutive model involving evolution laws. This procedure is suited for parameter identification of a given constitutive model. In addition, it may be used for neural network training (*i.e.* finding parameters \mathbf{p} containing weights and biases (\mathbf{W}, \mathbf{b})) when the constitutive model is described by a neural network.

Representing a constitutive model by a neural network enables to release of the form of the constitutive relation: potentials ψ and φ are searched in the space of functions that satisfy physical requirements defined in Section 2.1. Yet several questions and difficulties emerge when the constitutive model is described by a neural network. On the one hand, the question of consistency in the inference phase with respect to physical requirements is addressed in Section 2.1. On the other hand, the important number of parameters to find compared to the case of a given constitutive model form makes the optimization task

(Glorot & Bengio, 2010) more difficult. Specifically and as discussed in Chapter 2, Deep Learning is known to be sensitive to user-defined hyperparameters such as learning rate, number of epochs, batch size, etc. In Section 2.2, the automatic tuning strategy is discussed again and adapted (when necessary) to the minimization involving evolution laws.

2.1 Constitutive model described by a thermodynamically-consistent neural network

In Section 1.5.1, a thermodynamic framework has been defined for constitutive modeling. In the following, neural networks which satisfy this framework are presented. As a reminder, the constitutive behavior is described by means of two potentials: the Helmholtz free energy ψ and the dissipation pseudo-potential φ . To automatically satisfy thermodynamic principles, it is sufficient to assume ψ to be convex and φ to be convex, non-negative and zero at the origin.

According to the reliability of the initial guess on the constitutive model, either the whole behavior can be described by neural networks, or only some parts, as it is done in (Fuhg et al., 2023) in a context of limited data. In the case where some parts are well-known - such as elasticity - it is possible to enforce the form of the known part of the model. For example, function ψ can be split into $\psi = \psi_e(\epsilon_e) + \psi_p(\epsilon_p)$, in which $\psi_e = \frac{1}{2}\epsilon_e : \mathbf{K} : \epsilon_e$ and ψ_p is represented by a neural network. In this example, the proposed mCRE framework can simultaneously identify parameters of ψ_e and the weights and biases of the neural network describing ψ_p .

In both cases, it is possible to enforce positivity, convexity, and zero at origin constraints in the neural network. The general idea is described in the following and details can be found in (Fuhg et al., 2023; Linden et al., 2023). Positivity is enforced through the use of positive activation functions. Concerning convexity, the input convex neural network (ICNN) architecture proposed in (Amos et al., 2017) is employed. This architecture uses convex non-decreasing activation functions and positivity constraints on intermediate weights. As the composition of a convex and convex non-decreasing function is convex and the sum of convex functions is also convex, this architecture guarantees convexity. Finally, the output of the network is corrected by subtracting its value in zero, so that the output is zero when the input is zero.

2.2 Hyperparameters automatic tuning strategy

In this section, the strategy for automatic tuning of the hyperparameters is presented for the minimization involving evolution laws. The relevance of these rules is evaluated in Section 3. All the rules have been designed on a test case (with different target models and geometry which are not presented here) and evaluated on the test cases presented in the paper. The main changes concern the tuning of the weighting between losses and the tuning of the learning rate. The initialization strategy as well as the definition of the stopping criterion remain similar to the ones in Chapter 2 (see Sections 3.1 and 3.4). The following thus details

the Morozov-based automatic tuning of the weighting between losses, and the empirical adaptive learning rate rule. Figure 3.4 provides an idea of the general strategy for updating hyperparameters during training.

2.2.1 Morozov-based criterion for automatic tuning of weighting between losses

In the version of this work dedicated to state laws (Chapter 2), α was tuned at each epoch using a dichotomy (see Section 3.2). The computation time needed for the tuning of α was high (because it needs to re-perform Step 1 for each value of α tested) but still reasonable when dealing with state laws. On the contrary, when dealing with evolution laws, the computation time prohibits the use of this strategy. The idea here is to select an initial value of α , perform the training until convergence of the normalized CRE, and then progressively update the value of α (see Figure 3.4). As exploring the full range of possible values of α is prohibited by the computation time, the value of α is thus progressively increased (resp. decreased) if the model is updated above (resp. below) the noise level until one of the following conditions is met:

- the Morozov criterion is satisfied;
- or the normalized CRE is far from the user-defined target (the notion of far is defined through the use of the normalized CRE term to assess the modeling error).

After updating the value of α , Step 1 is initialized with the solution \hat{s} obtained before the update.

2.2.2 Adaptive tuning of the learning rate

The tuning of the learning rate is very similar to the strategy implemented in Chapter 2, Section 3.3. Evaluating the appropriate learning rate still involves an analysis of the progress achieved during a single training step. In this context, a valuable indicator of the training update speed can be obtained from the two-step minimization procedure. The number of iterations made in Step 1 for a given epoch is directly linked to the value of the update made in Step 2 of previous epochs, as illustrated in Figure 3.3. The difference with Chapter 2 is in the way to count the iterations, here the global iterations of the Step 1 fixed-point algorithm are counted. Starting from the second epoch, the learning is updated so that the number of iterations made is close to an empirically defined target on the number of iterations. This rule has been designed on a test case different from the one presented in this article and the influence of this empirical rule will be shown in Section 3.2.5. This rule is strongly very close to the one used in the previous chapter even though the minimization performed in Step 1 is different (so the way to count iterations is different).

According to multiple experiments, a good compromise is located around 6 iterations in Step 1. It is thus possible to automatically adapt the learning rate with the following empirical-based rule:

$$lr \leftarrow lr \times \text{update_coefficient}(\text{number_iterations_step_1}) \quad (3.15)$$

where $\text{update_coefficient}(\text{number_iterations_step_1})$ is a function depending on the number of iterations performed in previous Step 1. A point of attention is that the function $\text{update_coefficient}(\text{number_iterations_step_1})$ should be changed if the tolerance of the stopping criterion of Step 1 is modified (with a smaller tolerance the number of iterations is naturally higher). The main advantage of this rule is that the convergence of the method is no more sensitive to the user learning rate choice, as shown in Section 3.2.5, which enables to train the network online.

Near convergence, the learning rate should not be increased, as otherwise there is a risk of getting far from the global minimum. To do so, near convergence the learning rate is only updated if the rule tends to decrease the learning rate (the notion of near convergence is defined in the next paragraph). Additionally, a replay strategy has been implemented if the learning rate is too large, thus implying a very large number of iterations in Step 1. In this case, the model before updating is reloaded and a new gradient descent step is performed with a lower learning rate.

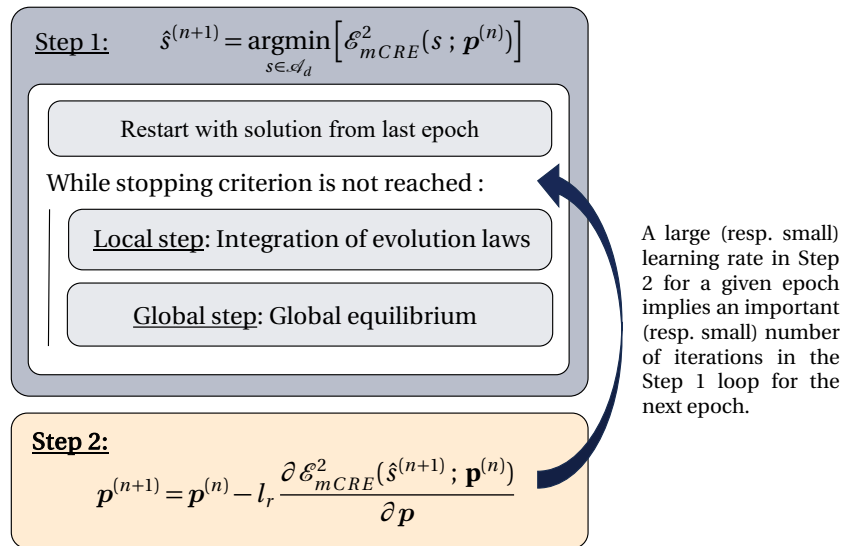


Figure 3.3 • Influence of the learning rate on the optimization process.

3 Results

This section is dedicated to the results and discussion on three different test cases. The first one in Section 3.1 is a toy example in 1D to illustrate the mCRE method in a case of parameter identification: only one parameter of a given hardening law is identified. The second one in Section 3.2 aims at learning a nonlinear hardening law. Finally the third one in Section 3.3 aims at learning a viscoplastic behavior. The evaluation of the performance is based on several criteria such as the accuracy of the learned model (Sections 3.2.3 and 3.3.2), the evolution of the loss function during the training (Sections 3.2.3 and 3.3.2), the noise robustness (Section 3.2.4), the localization of modeling error (Section 3.3.3), or the relevance of automatic hyperparameters tuning rules (learning rate in Section 3.2.5, and

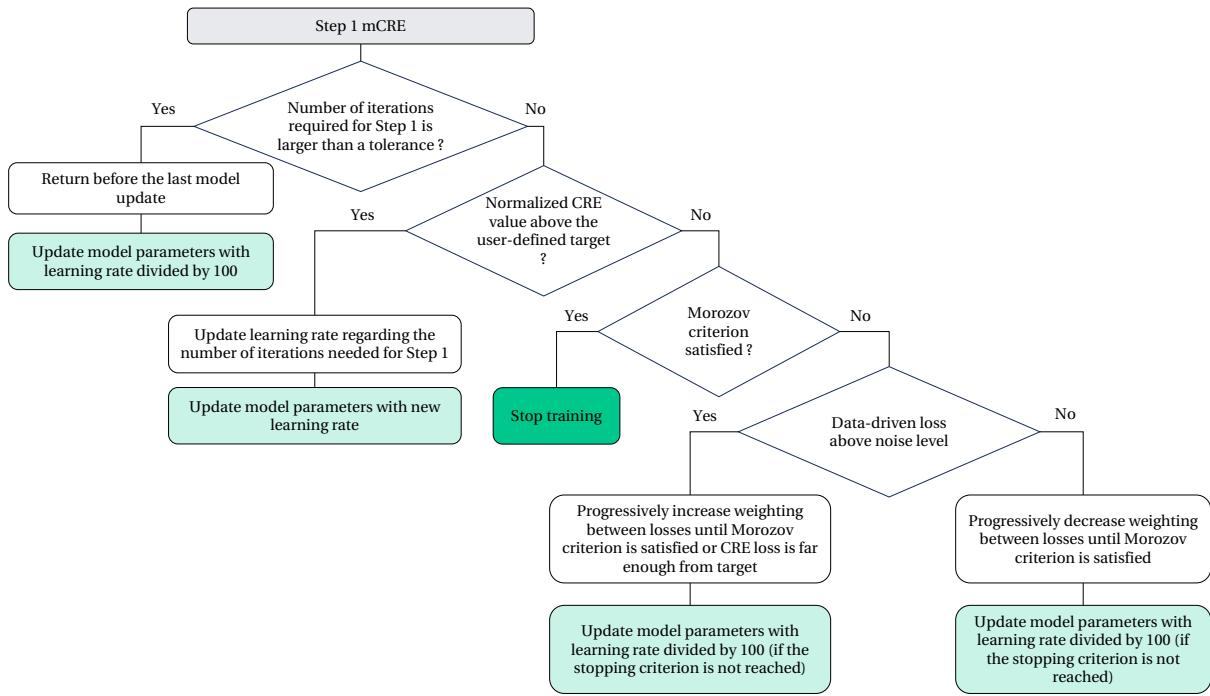


Figure 3.4 • General idea for hyperparameters updating.

the weighting between losses in Section 3.3.4).

3.1 A simple 1D example to identify an isotropic hardening modulus

Before presenting the results of neural network training with the mCRE framework, this section aims to show parameter identification of the hardening modulus in a simple 1D toy problem. This section is associated with an open-access code ¹ and can be helpful for a better understanding of the mCRE concept, hyperparameters influence and tuning. Figure 3.5 illustrates the problem that is addressed. A one-dimensional beam, with Young modulus E , initial elasticity limit R_0 , linear hardening modulus h and length $L = 1$, is loaded in tension. For the sake of simplicity, this problem is limited to one degree of freedom, in which the displacement is observed and affected by measurement noise. Here the constitutive laws are assumed to be known and the only parameter to identify is the hardening modulus h . The potentials are the following:

$$\psi(\epsilon_e, p) = \frac{1}{2} E \epsilon_e^2 + \frac{1}{2} h p^2 \quad (3.16)$$

$$\varphi^*(\sigma, R) = \begin{cases} 0 & \text{if } f < 0 \\ +\infty & \text{if } f = 0 \end{cases} \quad (3.17)$$

with $f = \sigma - (R + R_0)$.

¹https://gitlab-research.centralesupelec.fr/antoine.benady/mcre_evolution_1d/-/tree/main?ref_type=heads

Consequently,

$$\sigma = E \epsilon_e \quad (3.18)$$

$$R = h p \text{ with } p = \int_0^t \|\dot{\epsilon}_p\| dt \quad (3.19)$$

and, following the normality rule:

$$\dot{\epsilon}_p = \lambda \frac{\partial f}{\partial \sigma} \quad (3.20)$$

with $\lambda > 0$ if $f = 0$ and $\dot{f} = 0$.

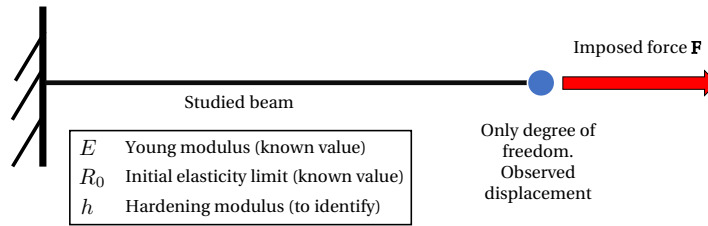


Figure 3.5 • Simple 1D problem: beam with isotropic hardening.

The initial guess is $h_{ini} = 5 \times 10^4$ MPa, whereas the target value is $h_{true} = 10^5$ MPa. Figure 3.6 shows the difference between true and predicted responses for the initial parameter guess. For a given training epoch, after the first step (computing an admissible solution that minimizes \mathcal{E}_{mCRE} for given model parameters), the expression of stationarity conditions of the Lagrangian defined in (3.13) gives that:

- s_u is a compromise between the solution of the forward problem and the observed data (hybrid state);
- s_v is the solution of the forward problem.

Figure 3.7 shows these solutions at the end of Step 1 for the first epoch, for different values of α . Indeed, the compromise between the forward problem and the observed data depends on the value of α . The larger the α , the closer the predicted displacement field to the true displacement field. In the example of Figure 3.7, two values of α are presented. With $\alpha = 10^6$, the predicted strain associated with solution s_u corresponds exactly to the true strain.

As mentioned in Section 2.2, the tuning of this parameter is important for at least two reasons:

- At the end of the training process, the compromise between the model and measurements should be such that the discrepancy between the predicted solution s_u and the observations is of the order of the magnitude of the noise level.
- As the gradient of the mCRE is directly linked to the difference between s_u and s_v , the value of α has a strong influence on the optimization process. Yet, this influence is strongly linked to the role of the learning rate. As tuning α with respect to the Morozov criterion is more computationally expensive than tuning the learning rate following the empirical rule presented in Section 2.2, this value is only tuned at convergence.

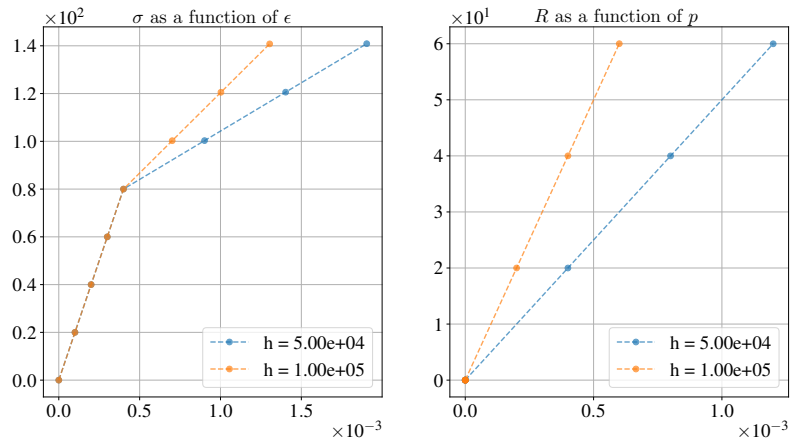


Figure 3.6 • True and predicted responses with initial parameters.

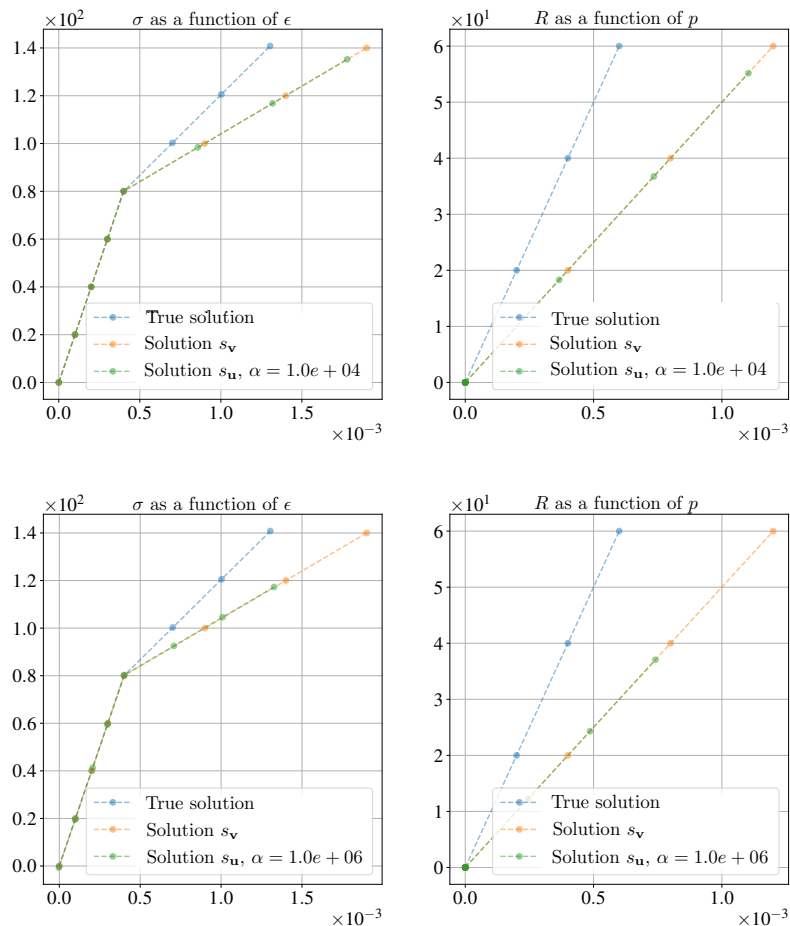


Figure 3.7 • Solutions s_u and s_v for different values of α (top: $\alpha = 10^4$, bottom $\alpha = 10^6$)

Figure 3.8 (top) shows the results of the training for various epochs. At the end of the training, the parameter h is properly identified and the solution s_v is close to the measurements. The constitutive relation, even though it is not directly observed, is properly learned. The bottom of Figure 3.8 shows that the mCRE functional decreases smoothly. In

this toy example, the hyperparameters are manually tuned because the computational time is very low (less than 30 seconds for the whole training).

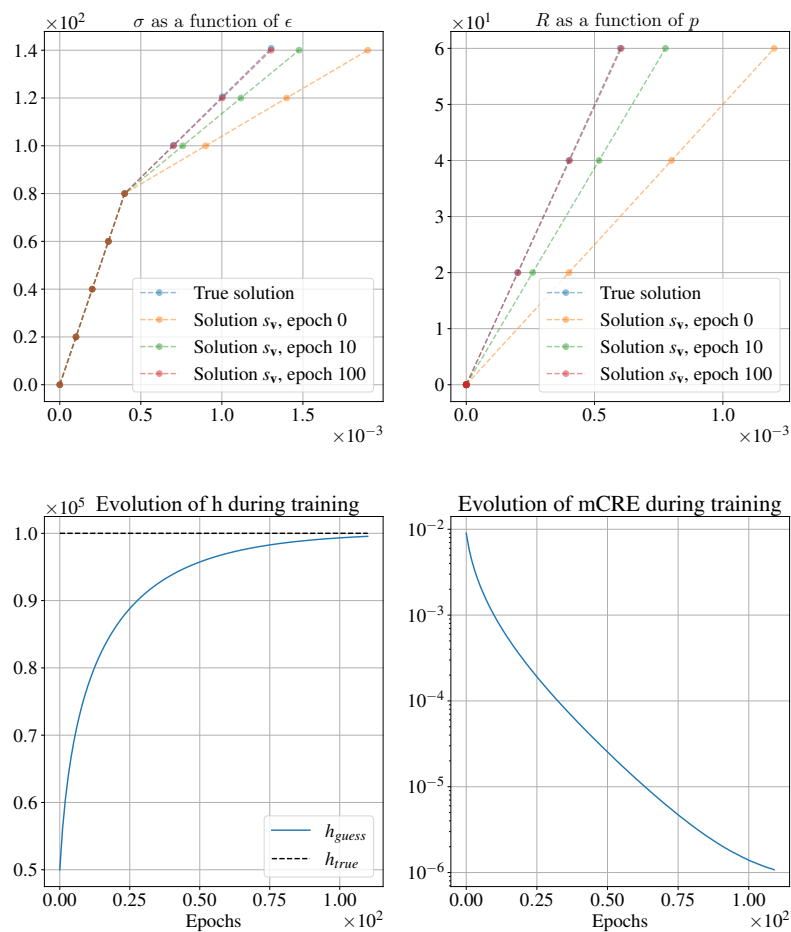



Figure 3.8 • Results of training for the identification of h for the 1D test case.

 **Online tool:** The code associated with this example is available in the GitHub repository in the file `Chap_3_evolution_mCRE.ipynb`

3.2 Rate-independent test case: learning a nonlinear isotropic hardening law

3.2.1 Reference problem

In the previous section, the model form was known and only one parameter was identified. In the use cases targeted in this paper, only an initial guess on the model form is known. The initial guess is represented by a neural network that is trained in the mCRE framework. To represent the initial guess, it is possible to perform a first supervised training with the Adam optimizer for example (Kingma & Ba, 2015) before the mCRE training.

In this section, a nonlinear isotropic hardening law is recovered. For the sake of simplicity, a part of the model is assumed to be known as represented in Table 3.1, with the values of the parameters in Table 3.4. This is a modeling choice dictated by the reliability of material knowledge: other strategies can be imagined such as identifying some model parameters simultaneously or even representing every part with neural networks. The mCRE strategy easily adapts to these cases. It is worth noticing that in the addressed problem, the yield criterion is assumed to be known, which means that the dissipation potential is known (it is recalled that in the rate-independent case, the pseudo-potential is the indicator function of the convex elastic domain as defined in (1.30)). In the rate-independent case, representing the dissipation potential by a neural network is not convenient: it is more adapted to use the neural networks to represent the yield criterion.

Figure 3.9 shows the geometry, boundary conditions, and sensor positions of the problem under study. The studied case is a 2D beam loaded in tension with variable loading forces. The generated measurements come from two optic fibers oriented along the two main axes and positioned in the middle of the beam.

| Model used to generate data | Model to train |
|--|---|
| Elastic part of the free energy: | |
| $\psi_e(\epsilon_e) = \frac{1}{2}(\lambda(\text{tr } \epsilon_e)^2 + 2\mu\epsilon_e : \epsilon_e)$ | Assumed to be known with the correct parameters |
| Plastic part of the free energy: | |
| $\psi_p(\epsilon_p) = Ap + \frac{A}{B}(e^{-Bp} - 1)$ | Represented by a neural network initialized with: $\psi_p(\epsilon_p) = \frac{1}{2}hp^2$ |
| Limit of the elasticity domain: | |
| $f = \sigma_{eq} - (R + R_0)$ | Assumed to be known with the correct parameters |

Table 3.1 • Summary of the nonlinear hardening law test case, with $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$, $\mu = \frac{E}{2(1+\nu)}$ and σ_{eq} the Von-Mises equivalent stress.

| Parameters | Value |
|------------|---------|
| E | 200 GPa |
| ν | 0.25 |
| A | 45 MPa |
| B | 3000 |
| h | 50 GPa |
| R_0 | 100 MPa |

Table 3.2 • Values of parameters used in the nonlinear hardening law test case.

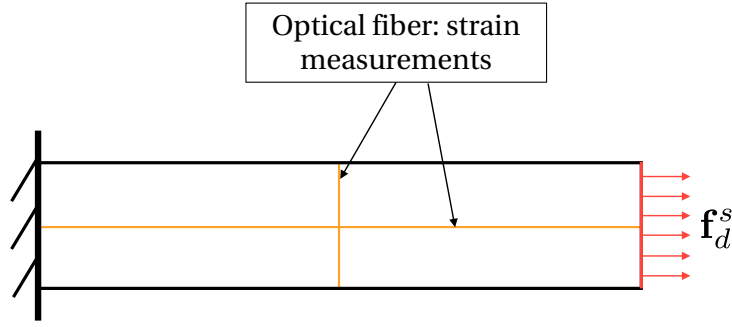


Figure 3.9 • Geometry of the problem under study.

3.2.2 mCRE minimization for the rate-independent test case.

This section aims to detail the mCRE minimization in the specific case treated. Step 1 minimization of the mCRE is composed of a local step (integration of evolution laws) and a global step (computation of admissible fields).

Local step

In the local step, the variables $\epsilon(\mathbf{u})$ and $\epsilon(\mathbf{v})$ and the variables $p_u, p_v, R_u, R_v, \epsilon_{e,u}, \epsilon_{e,v}, \epsilon_{p,u}, \epsilon_{p,v}$ are searched. The integration is performed with an Euler implicit scheme. In the following, the indices \mathbf{u} and \mathbf{v} are dropped and the integration is written in a generic form. For each element and for each time step, the integration consists in finding (both for \mathbf{u} and \mathbf{v} solutions):

$$\begin{cases} \dot{\epsilon}_p = \dot{\lambda} \frac{\partial f}{\partial \sigma} \\ \dot{p} = -\dot{\lambda} \frac{\partial f}{\partial R} \\ f = 0 \\ R = \frac{\partial \psi_p(p)}{\partial p} \\ \dot{\epsilon} = \dot{\epsilon}_e + \dot{\epsilon}_p \\ \sigma = \frac{\partial \psi_e}{\partial \epsilon_e} \end{cases} \quad (3.21)$$

After time discretization, (3.21) becomes:

$$\begin{bmatrix} \Delta \epsilon - (\epsilon_e^{t+1} - \epsilon_e^t) - \Delta \lambda \frac{\partial f}{\partial \sigma} \\ p^{t+1} - p^t + \Delta \lambda \frac{\partial f}{\partial R} \\ R^{t+1} - R^t + \Delta \lambda \frac{\partial f}{\partial R} \frac{\partial^2 \psi_p(p)}{\partial p^2} \\ f(\epsilon_e^{t+1}, R) \\ \sigma - \frac{\partial \psi_e}{\partial \epsilon_e} \end{bmatrix} = \mathbf{0} \quad (3.22)$$

where \cdot^t denotes the quantity \cdot at time step t . The system (3.27) is solved with a Newton-Raphson method in which the Jacobian matrix is computed through automatic differentia-

tion. At the end of the local step, the variables $p_u, p_v, R_u, R_v, \epsilon_{e,u}, \epsilon_{e,v}, \epsilon_{p,u}, \epsilon_{p,v}$ are updated.

Global step

During the global step, the variables $X_u, Y_u, \epsilon_{p,u}, X_v, Y_v, \epsilon_{p,v}$ are frozen to the value obtained at the local step. An admissible solution is computed through the minimization of the Lagrangian defined in (3.13), with $X_u = r_u, Y_u = R_u, X_v = r_v, Y_v = R_v$

Step 2

As the dissipation potential is assumed to be known here, only the parameters involved in the plastic part of the free energy \mathbf{p}_{ψ_p} are updated through gradient descent:

$$\mathbf{p}_{\psi_p}^{n+1} = \mathbf{p}_{\psi_p}^n - \frac{\partial \mathcal{E}_{mCRE}^2(s_u, s_v; \mathbf{p})}{\partial \mathbf{p}_{\psi_p}} \quad (3.23)$$

with

$$\frac{d \mathcal{E}_{mCRE}^2(s_u, s_v; \mathbf{p})}{d \mathbf{p}_{\psi_p}} = \int_0^T \int_{\Omega} \left(\frac{\partial \psi}{\partial \mathbf{p}_{\psi_p}} \Big|_{s_u} - \frac{\partial \psi}{\partial \mathbf{p}_{\psi_p}} \Big|_{s_v} \right) d\Omega dt \quad (3.24)$$

3.2.3 General results

At the beginning of the training, the initial guess on the model form is wrong (see Table 3.1), which is referred to as model bias. If the form of the model were not released (*i.e.* not described by a neural network), it would not be possible to properly fit observations. In Figure 3.10, the material behavior is shown for the true model and initial guess. On the right, the shape of the hardening part of σ_{11} as a function of ϵ_{11} is not properly represented. The initial guess is linear whereas the true response has an exponential shape.

As previously mentioned, representing a constitutive model by a neural network enables relaxing the model form. Indeed, Figure 3.11 shows the response at the end of the training: the hardening law has evolved from a linear law to a nonlinear law and the response is properly represented. The relation R as a function of p is learned even though none of these quantities is observed in practice. Note that the case of non-monotonic loadings has not been addressed in this article, but has been treated previously in the case of parameter identification (see Chapter 5 in (Nguyen, 2021)). As the fixed-point algorithm is global in time, non-monotonic loadings should not have any consequence on the convergence.

Figure 3.12 shows the evolution of the mCRE (with the CRE and distance to observation term) during the training. At the end of the training, the normalized data loss is approximately 1, which means that the Morozov criterion is satisfied. Concerning the normalized CRE, it is below the user-defined target value. As these two criteria are met and following the procedure defined in Figure 3.4, the training stops. The gap between the learned relation and the true one (see Figure 3.11) is explained by the Morozov criterion: the updated model

should not fit data below the noise level (1% here, where the noise is on the total strain observations coming for the optic fiber measurements).

After epoch 500, the normalized CRE has an oscillating behavior, which is explained by the updating of the weighting between losses α . The evolution of the parameter α is represented in Figure 3.13. The high value of the loss function can be surprising compared to value of other loss functions in the literature. This is explained by the high value of α (20,000 in Figure 3.13) needed to satisfy the Morozov criterion, as well as the fact that the CRE term is not normalized in the value of the mCRE plotted.

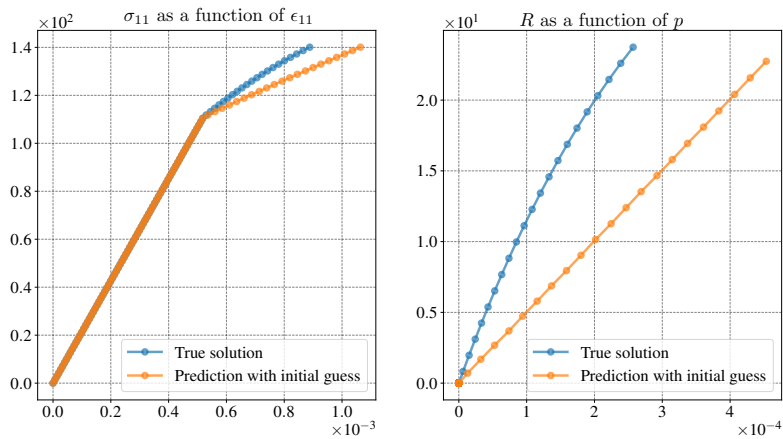


Figure 3.10 • Model bias at the beginning of the training.

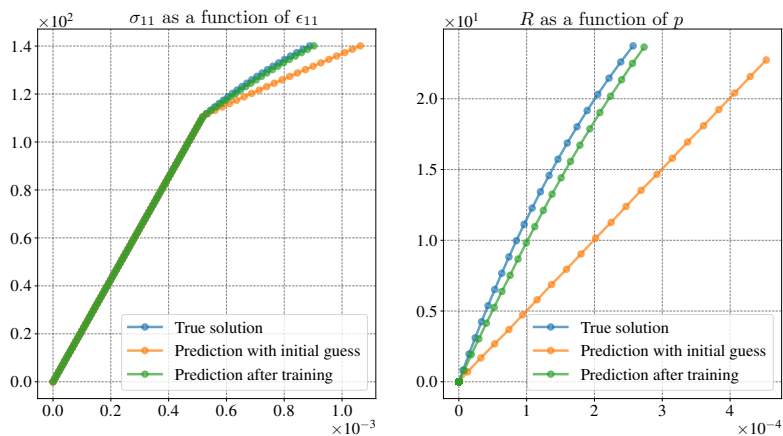


Figure 3.11 • Learned hardening law at the end of the training.

3.2.4 Robustness to noise level

Previous works about mCRE have shown the high robustness of the method to noise level (see (Allix et al., 2005; Feissel & Allix, 2007) for example). This section aims to briefly analyze the noise robustness of this test case. In Figure 3.14, the evolution of the loss function, divided into the CRE and the distance to observations, is represented for different noise

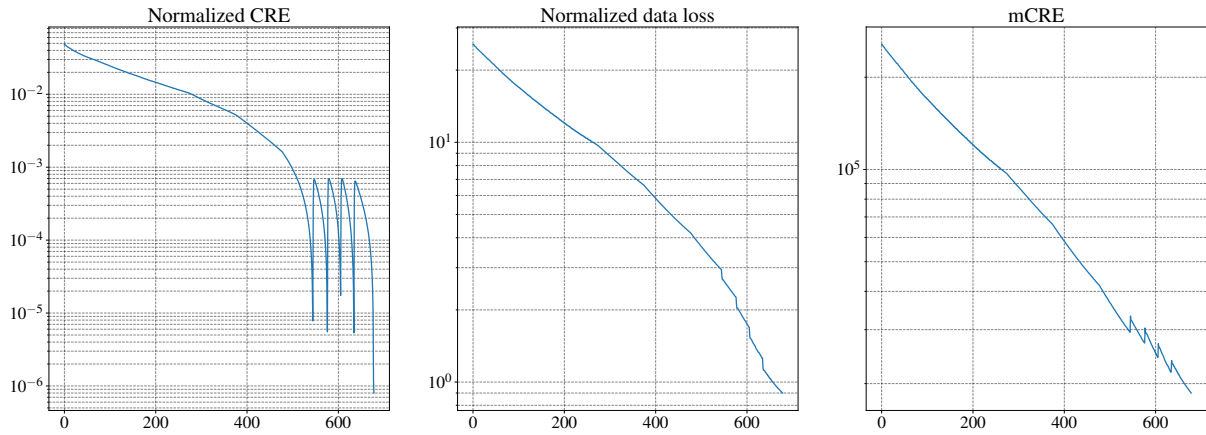


Figure 3.12 • Left: evolution of the normalized CRE during the training. Middle: evolution of the normalized data loss during the training. Right: evolution of the mCRE during the training. For the 3 curves, the values are computed after Step 1 and before updating the weights and biases of neural networks.

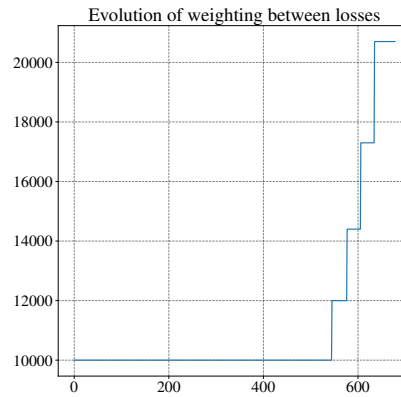


Figure 3.13 • Evolution of α during the training.

levels from 0.01% to 20%. The first observation to notice is the value of the normalized data loss at the end of the training. This value is close to one, which means that the Morozov criterion is satisfied: the gap between the predicted strain and the measured strain is the same order of magnitude as the noise level. Secondly, the normalized CRE decreases during training, until the stopping criterion is met. These two observations show that the training is reaching the expected behavior: the modeling error is decreased to the expected value and the model is updated up to noise level.

In the following, a $X\%$ noise is added to the measurements, following a normal distribution of zero mean and a standard deviation η defined as:

$$\eta = X\% \times \frac{\sum |u_{obs}|}{\text{number of observed degrees of freedom}} \quad (3.25)$$

When the noise is 20%, the right part of Figure 3.14 shows that the mCRE is increasing while the normalized CRE and the normalized data loss are decreasing. This is explained by the increasing value of α that follows an evolution similar to the one in Figure 3.13. For

the highest noise level, the training requires fewer epochs than the lower noise level. The first reason is that the value of α at the beginning of the training is adequate (normalized data loss is close to one) and as the model should not be updated below the noise level, the correction from the initial guess is lower than for lower noise levels as it is further shown.

Now that the behavior of the training has been checked, it is interesting to observe the curve of the learned hardening law and the response σ_{11} as a function of ϵ_{11} . Figure 3.15 shows this for two different noise levels (0,01% and 20%). For the low noise level (even though it is a quite high noise level compared to other methods such as (Thakolkaran et al., 2022)) the hardening law is properly learned. On the other hand, for the noise level of 20%, only a slight correction has been made. Indeed, on the left of Figure 3.15, the gap between the predicted strain and the true strain (without noise on the curve) is around 20%: it would make no sense to update below the noise level.

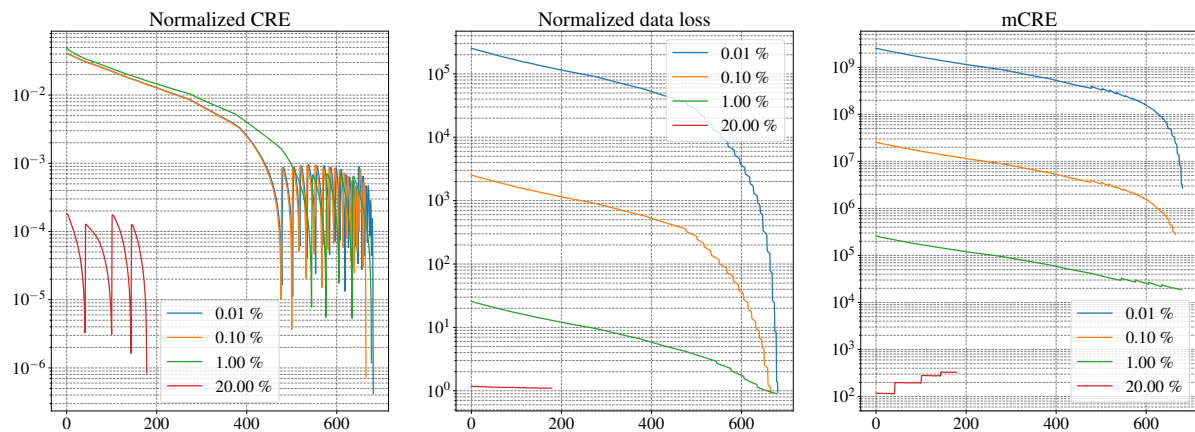


Figure 3.14 • Impact of noise level on convergence. Left: evolution of the normalized CRE during the training. Middle: evolution of the normalized data loss during the training. Right: evolution of the mCRE during the training. For the 3 curves, the values are computed after Step 1 and before updating the weights and biases of neural networks.

3.2.5 Relevance of the automatic learning rate tuning

This section aims to briefly illustrate the efficiency of the adaptive tuning of the learning rate presented in Section 2.2. In Figure 3.16, the evolution of the mCRE during training is represented for three different values of the initial learning rate chosen by the user (with a difference of 5 orders of magnitude between the highest and the lowest). One strong advantage of this rule is that the level of mCRE reached at convergence is the same for the 3 initial values. This means that the training is not sensitive to the user learning choice, which is rarely the case in gradient-descent-based optimization in Deep Learning (Goodfellow et al., 2016). This is explained by the adaptive rule which automatically chooses close values of the learning rate after several epochs, independently of the initial choice. A sensitivity remains regarding the training time, which is not a major concern here.

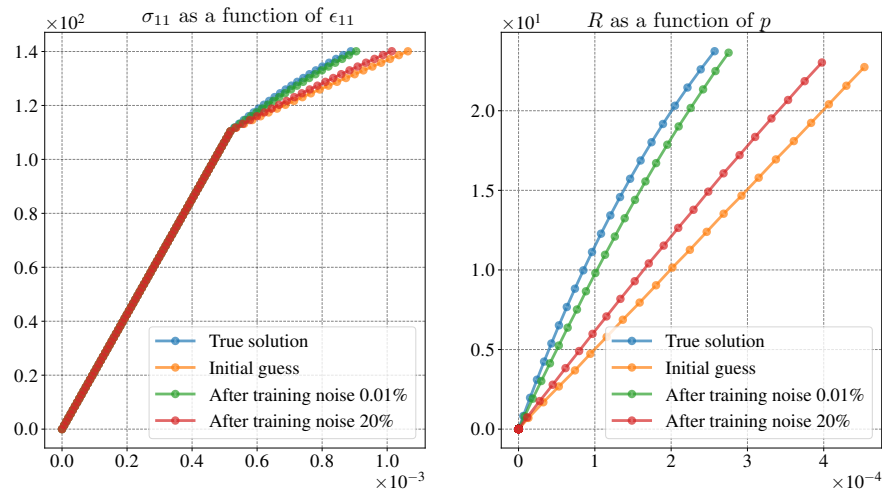


Figure 3.15 • Learned hardening law at the end of training for different noise levels.

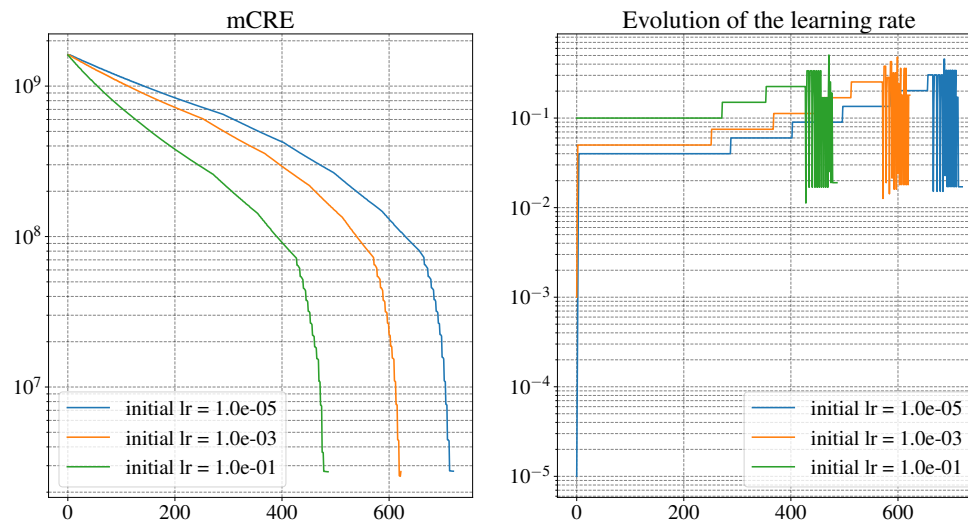


Figure 3.16 • Influence of the initial learning rate on the optimization process.

3.3 Rate-dependent test case: learning a dissipation pseudo-potential

In this section, the method is tested with a viscoplastic behavior. As in the previous section, a part of the constitutive model is assumed to be known and another part is released through the use of a neural network. Here, the free energy is assumed to be fully known (linear elasticity and linear hardening law), as well as the boundary of the elasticity domain. The parameters involved in the free energy and elasticity domain are also assumed to be known (even though they could be updated in the same way as the neural network parameters). In contrast, the form of the dissipation pseudo-potential is released with a neural network. This neural network is initialized to represent a classical power law (Lemaitre & Chaboche, 1990), whereas the model used to generate the data is a hyperbolic cosine law (Pipard et al.,

2013), as summarized in Table 3.3. Figure 3.18 shows the discrepancy between the initial and the true models.

The case under study is the 2D notched beam shown in Figure 3.17. The beam is subjected to tension with varying loading forces, while the full-field displacement is measured.

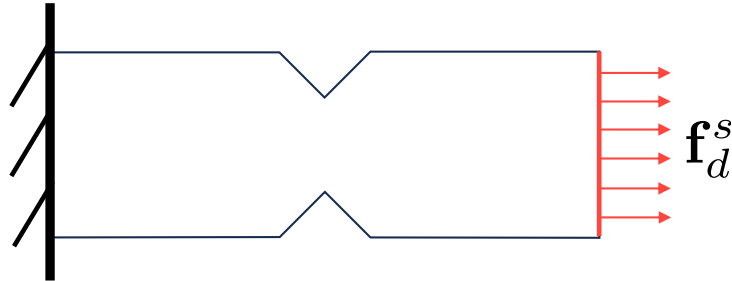


Figure 3.17 • Geometry of the problem under study.

| Model used to generate data | Model to train |
|--|--|
| Elastic part of the free energy: | |
| $\psi_e(\epsilon_e) = \frac{1}{2}(\lambda(\text{tr } \epsilon_e)^2 + 2\mu\epsilon_e : \epsilon_e)$ | Assumed to be known with the correct parameters |
| Plastic part of the free energy: | |
| $\psi_p(\epsilon_p) = \frac{1}{2}h p^2$ | Assumed to be known with the correct parameters |
| Limit of the elasticity domain: | |
| $f = \sigma_{eq} - (R + R_0)$ | Assumed to be known with the correct parameters |
| Dissipation potential | |
| $\varphi^*(f) = AB\left(\cos\left(\frac{\langle f \rangle_+}{A}\right) - 1\right)$ | Represented by a neural network initialized with: $\varphi^*(f) = \frac{K_a}{N_a + 1} \left(\frac{\langle f \rangle_+}{K_a}\right)^{N_a + 1}$ |

Table 3.3 • Summary of the rate-dependent test case.

| Parameters | Value |
|------------|----------------------------------|
| E | 200 GPa |
| ν | 0.3 |
| A | 18 MPa |
| B | $0.3s^{-1}$ |
| h | 1.5 GPa |
| R_0 | 160 MPa |
| K_a | $50 \text{ MPa} \cdot s^{1/N_a}$ |
| N_a | 10 |

Table 3.4 • Values of parameters used in the rate-dependent test case.

3.3.1 mCRE minimization for the rate-dependent test case

This section aims to detail the mCRE minimization in the case of a rate-dependent behavior. Step 1 minimization of the mCRE is composed of a local step (integration of evolution laws) and a global step (computation of admissible fields).

Local step

As in the rate-independent case, in the local step, the variables $\epsilon(\mathbf{u})$ and $\epsilon(\mathbf{v})$ and the variables $p_u, p_v, R_u, R_v, \epsilon_{e,u}, \epsilon_{e,v}, \epsilon_{p,u}, \epsilon_{p,v}$ are searched. The integration is performed with an Euler implicit scheme. In the following, the indices \mathbf{u} and \mathbf{v} are dropped and the integration is written in a generic form. For each element and for each time step, the integration consists in finding (both for \mathbf{u} and \mathbf{v} solutions):

$$\begin{cases} \dot{\epsilon}_p = \frac{\partial \varphi^*}{\partial \boldsymbol{\sigma}} \\ \dot{p} = \frac{\partial \varphi^*}{\partial R} \\ R = \frac{\partial \psi_p(p)}{\partial p} \\ \dot{\epsilon} = \dot{\epsilon}_e + \dot{\epsilon}_p \\ \boldsymbol{\sigma} = \frac{\partial \psi_e}{\partial \epsilon_e} \end{cases} \quad (3.26)$$

After time discretization, (3.26) becomes:

$$\begin{bmatrix} \Delta \epsilon - (\epsilon_e^{t+1} - \epsilon_e^t) - \frac{\partial \varphi^*}{\partial \boldsymbol{\sigma}} dt \\ p^{t+1} - p^t + \frac{\partial \varphi^*}{\partial R} dt \\ R^{t+1} - R^t + \frac{\partial \varphi^*}{\partial R} \frac{\partial^2 \psi_p(p)}{\partial p^2} dt \\ \boldsymbol{\sigma} - \frac{\partial \psi_e}{\partial \epsilon_e} \end{bmatrix} = \mathbf{0} \quad (3.27)$$

where \cdot^t denotes the quantity \cdot at time step t , and dt the time increment. The system (3.27) is solved with a Newton-Raphson method in which the Jacobian matrix is computed through automatic differentiation. At the end of the local step, the variables $p_u, p_v, R_u, R_v, \epsilon_{e,u}, \epsilon_{e,v}, \epsilon_{p,u}, \epsilon_{p,v}$ are updated.

Global step

During the global step, the variables $\mathbf{X}_u, \mathbf{Y}_u, \epsilon_{p,u}, \mathbf{X}_v, \mathbf{Y}_v, \epsilon_{p,v}$ are frozen to the value obtained at the local step. An admissible solution is computed through the minimization of the Lagrangian defined in (3.13), with $\mathbf{X}_u = r_u, \mathbf{Y}_u = R_u, \mathbf{X}_v = r_v, \mathbf{Y}_v = R_v$

Step 2

As the free energy is assumed to be known here, only the parameters involved in the dissipation potential \mathbf{p}_{φ^*} are updated through gradient descent:

$$\mathbf{p}_{\varphi^*}^{n+1} = \mathbf{p}_{\varphi^*}^n - \frac{\partial \mathcal{E}_{mCRE}^2(s_u, s_v; \mathbf{p})}{\partial \mathbf{p}_{\varphi^*}} \quad (3.28)$$

with

$$\frac{\partial \mathcal{E}_{mCRE}^2(s_u, s_v; \mathbf{p})}{\partial \mathbf{p}_{\varphi^*}} = \int_0^T \int_0^t \int_{\Omega} \left(\frac{\partial \varphi^*}{\partial \mathbf{p}_{\varphi^*}} \Big|_{s_v} - \frac{\partial \varphi^*}{\partial \mathbf{p}_{\varphi^*}} \Big|_{s_u} \right) d\Omega ds dt \quad (3.29)$$

3.3.2 General results

This section presents the general results of the method on the viscoplastic test case. Figure 3.18 shows the results of the finite element simulation for the true model and neural network before and after training. The behavior is properly reconstructed after training and the model bias on the dissipation potential is properly corrected.

Figure 3.19 illustrates the evolution of the different terms of the mCRE during the training. At the end of the training, the stopping criterion is properly met as the normalized CRE is below the target ($1e^{-6}$) and the normalized data loss is close to 1, thus satisfying the Morozov criterion. The oscillating behavior of the normalized CRE is explained by the automatic update of the weighting between losses α . Indeed, every time the normalized CRE is below the target and the Morozov criterion is not met, the value of α is increased (see Figure 3.4). When α is increased, the normalized CRE increases and the normalized data loss decreases.

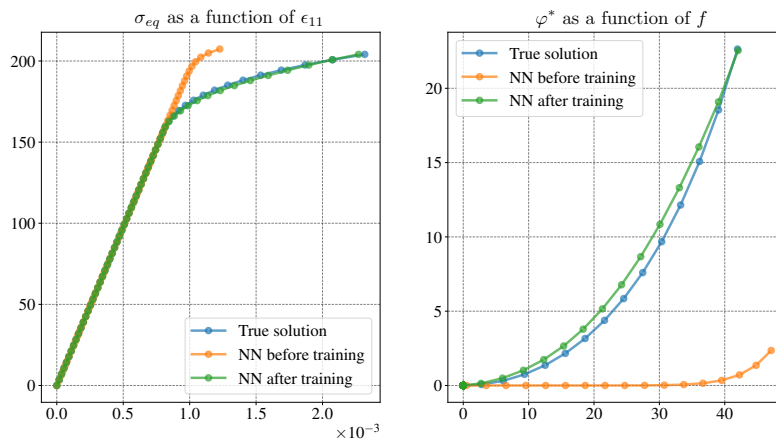


Figure 3.18 • Finite element simulation for the true model and neural network before and after training (with 0.1% noise level).

The previously obtained conclusions in the rate-independent test case regarding the robustness to noise level (Section 3.2.4) and the relevance of the learning rate tuning (Section 3.2.5) still stand in the rate-dependent test case (even though they are not presented again).

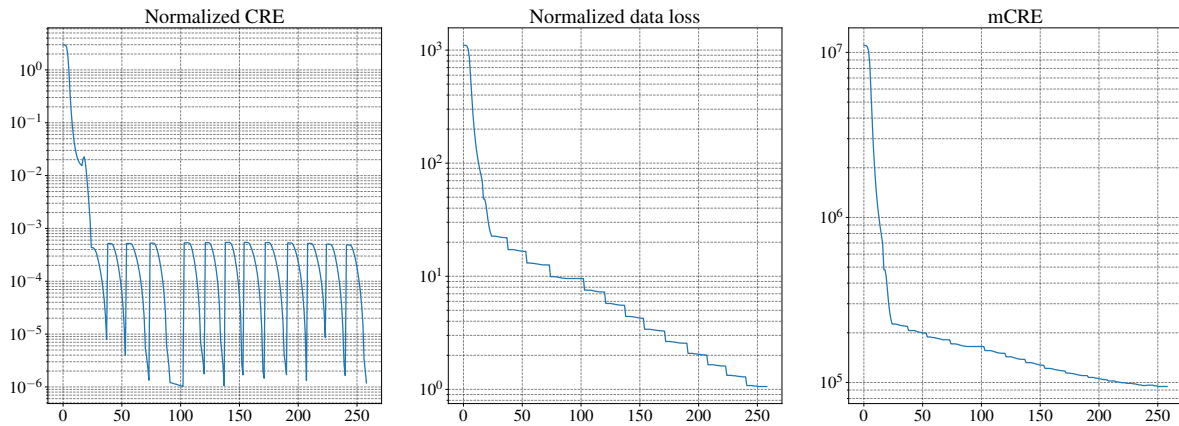


Figure 3.19 • Left: evolution of the normalized CRE during the training. Middle: evolution of the normalized data loss during the training. Right: evolution of the mCRE during the training.

The following sections evaluate the mCRE framework on the ability to properly localize the model bias on the structure in Section 3.3.3 and the relevance of the tuning of the weighting between losses α in Section 3.3.4.

3.3.3 Localization of the model bias in the structure

An interesting aspect of the mCRE framework is the localization of model bias. In the present example, Figure 3.18 shows that the plastic behavior is not properly modeled before the training of the neural network. This can also be seen in Figure 3.20 in which the cumulative plastic strain is shown in the structure for the last time step of the simulation, both for the true model and the neural network before training. On the right of Figure 3.20, the normalized CRE shows that the modeling error is localized in the part of the structure in which there is plasticity. This observation is explained by the fact that this is the plastic part of the behavior that is not properly modeled. It is worth noticing that the computation of this error only requires information that is available in the inference phase (no use of the true potential nor true internal variable value).

3.3.4 Relevance of the automatic tuning of the weighting between losses

This section aims to briefly illustrate the relevance of the automatic tuning of the weighting between losses presented in Section 2.2. In Figure 3.21, the evolution of the mCRE during training is represented for four different values of the initial weighting between losses chosen by the user (with a difference of 4 orders of magnitude between the highest and the lowest). Figure 3.4 can be helpful to understand the evolution of α during training: the value of α is increased every time the normalized CRE is below the target and the Morozov criterion is not met. Figure 3.21 shows that the level of mCRE reached at convergence is the same for the 4 initial values, which means that the converged value is not sensitive to the user's choice of the initial value. Yet, a strong sensitivity remains regarding the computation

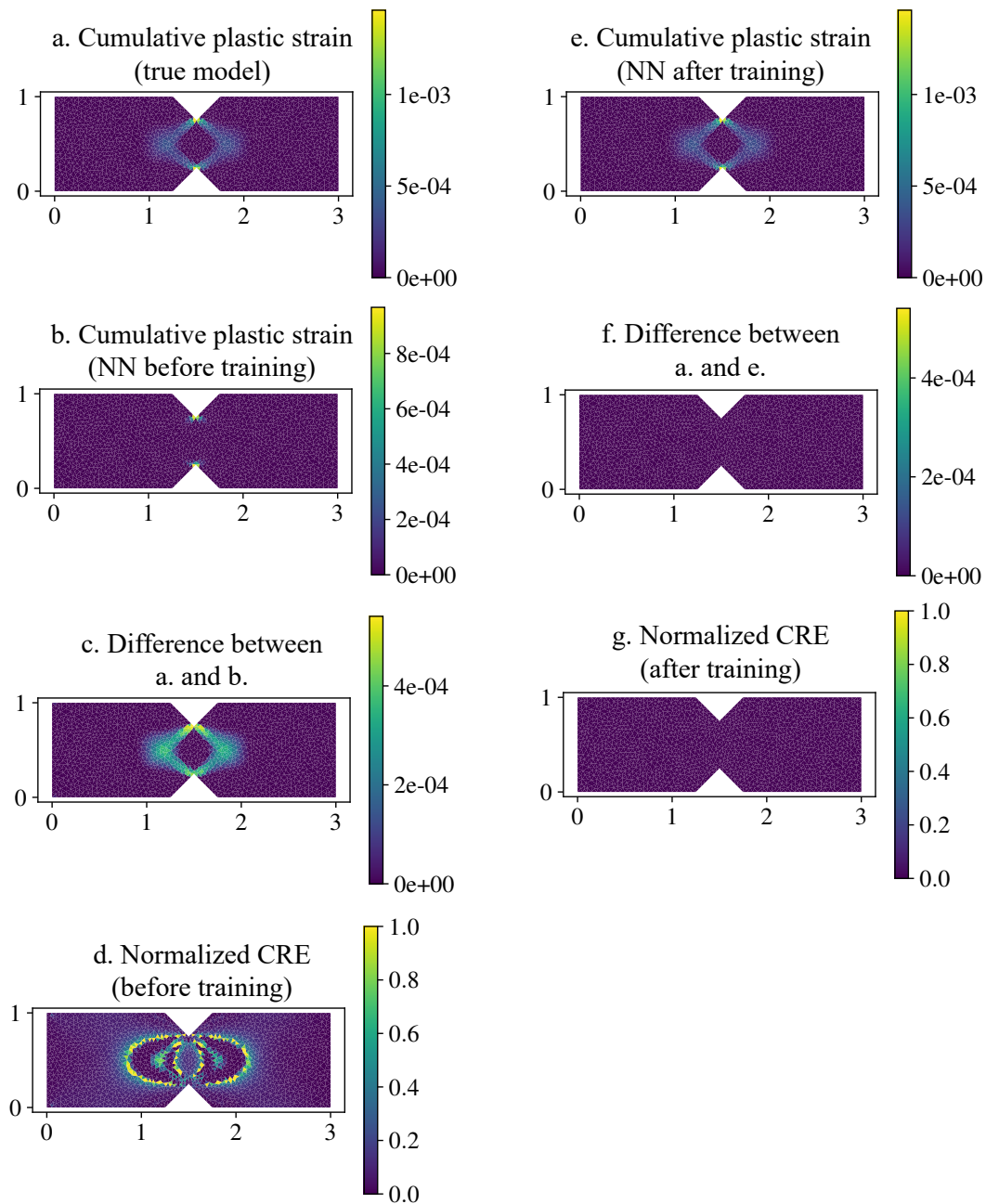


Figure 3.20 • a: Cumulative plastic strain obtained with the true model. b: Cumulative plastic strain obtained with the neural network before training. c: Difference between the cumulative plastic strain obtained with the true model and the cumulative plastic strain obtained with the neural network before training. d: Normalized CRE at the beginning of the training. e: Cumulative plastic strain obtained with the neural network after training. f: Difference between the cumulative plastic strain obtained with the true model and the cumulative plastic strain obtained with the neural network after training. g: Normalized CRE after training.

time. Indeed a proper initial value can help achieve quicker convergence ($\alpha = 10000$) in

Figure 3.21. This issue might be overcome by adding an initial automatic tuning at the beginning of the training, such as is done in the previous chapter.

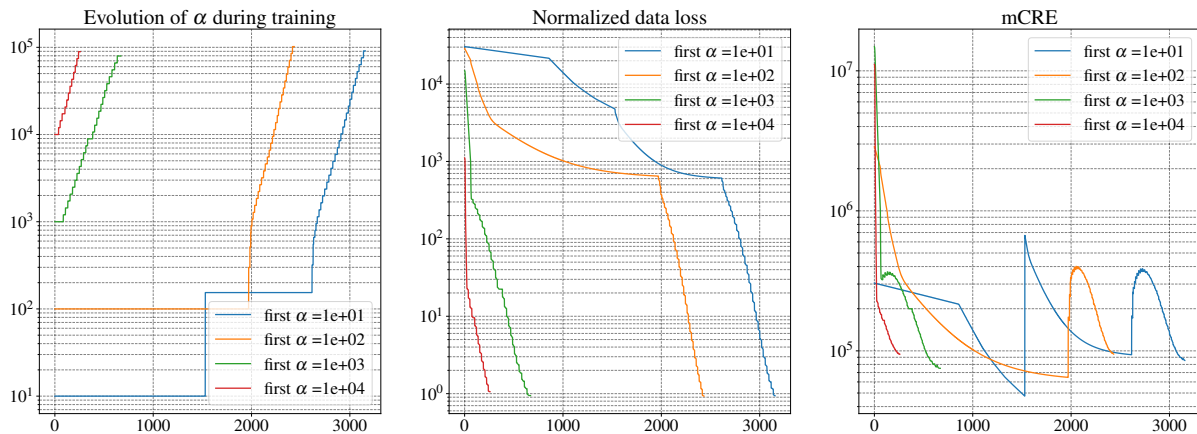


Figure 3.21 • Left: evolution of the weighting between losses during the training. Middle: evolution of the normalized data loss during the training. Right: evolution of the mCRE during the training.

Chapter conclusion

This chapter presented a method for unsupervised training physics-augmented neural networks to represent evolutionary laws. This chapter is built on the framework of Chapter 1, in which the modified Constitutive Relation Error is minimized. All reliable information is still enforced by construction. The novelty of this chapter lies in the ability to treat history-dependent dissipative behavior. This extension required the adaptation of the minimization process, especially in Step 1 (computation of admissible field) which requires the integration of the evolution laws.

The performance of the methods was assessed on three test cases: a simple 1D problem where only one parameter of the constitutive model was updated, an elastoplastic behavior where a nonlinear hardening law was learned, and a viscoplastic behavior where a cosine hyperbolic dissipation potential was learned. The method showed interesting performance in terms of accuracy of the learned model, robustness to noise, localization of modeling error, and sensitivity of the law to user-defined hyperparameters. Since the model should be corrected within the DDDAS paradigm, special attention was paid to the automatic tuning of the hyperparameters (weighting between losses, learning rate, and initialization). In summary, the proposed approach represents a promising tool for predicting the response of materials and structures to external loads.

However, this work requires additional studies before being applied to real structures. First, the computational cost seems prohibitive. This problem could be overcome by coupling this work with previous ones on the mCRE suitable for real-time control using reduced order modeling (Marchand et al., 2016; Chamoin et al., 2016). Another reason is the required choice of postulated internal variables. Even if the model form of the relationships involving internal variables is properly corrected, a model bias remains in the choice of internal variables. A research direction to alleviate this problem could be the use of recurrent neural networks (RNN), in which the internal memories can play the role of internal variables (Gorji et al., 2020), although the coupling with the mCRE error could be challenging.

4

On the need for an adequate parameterization of the neural networks

Contents

| | | |
|----------|---|------------|
| 1 | The problem of local minima with NN-mCRE | 92 |
| 1.1 | The specificities of the NN-mCRE with initialization | 92 |
| 1.2 | An example of initialization bias | 92 |
| 1.3 | Possibilities to get rid of initialization bias | 94 |
| 2 | An adequate parameterization of neural networks for constitutive model | 95 |
| 2.1 | A one-layer modified ICNN | 96 |
| 2.2 | Expressivity of one-layer modified ICNN | 99 |
| 2.3 | Interpretable extrapolation offered by one-layer modified ICNN . . . | 99 |
| 2.4 | Discussion on the use of second-order optimization method | 101 |
| 3 | Results with one-layer modified ICNN | 102 |
| 3.1 | Example with a nonlinear elastic behavior | 102 |
| 3.2 | Example of interpretable extrapolation in elastoplasticity | 103 |

So far, the focus of this manuscript has been on the development of an mCRE minimization procedure for unsupervised training of neural networks. However, a fundamental question has not yet been addressed at this stage, that of the architecture (number of layers and number of neurons per layer) of the neural network. This important question deserves a chapter of its own. As already mentioned, minimizing the mCRE functional requires a relevant initialization and if we are not careful enough, it is possible to introduce an initialization bias due to the large number of local minima. If the optimization gets stuck into a local minimum, the model bias cannot be corrected. This chapter details the problem of initialization bias and proposes a method for choosing the architecture to overcome this problem.

1 The problem of local minima with NN-mCRE

The training of neural networks is known to be sensitive to initialization (Glorot & Bengio, 2010) because of the strongly nonconvex nature of the loss function with respect to the neural networks parameter. The minimization of the mCRE is also sensitive to initialization, even in the case of parameter updating (Nguyen, 2021) (without neural network). Without being cautious enough, the integration of neural networks in the mCRE can strongly emphasize this sensitivity to initialization. In Chapters 2 and 3, a supervised initialization training was performed before the mCRE minimization. In the cases treated so far, this method was not subjected to any problem, but we show in the following that an initialization bias can appear. First, some specificities of the mCRE are recalled to justify the need for a close initialization of the neural network in Section 1.1. In Section 1.2, an example is introduced to illustrate the risk for an initialization bias which prevents properly correcting model bias. Finally, Section 1.3 discusses the possible solution to get rid of initialization bias.

1.1 The specificities of the NN-mCRE with initialization

The minimization procedure of the mCRE requires a relevant initialization for 2 reasons, and contrary to most methods in Deep Learning, a random initialization is not suitable. The first reason is that for each mCRE iteration, admissible fields are computed, using the current model. With a randomly initialized model, the fields obtained at the end of Step 1 would not be meaningful. The second reason can be understood when observing the modeling error term in the mCRE: it involves the potential described by a neural network. With a random potential, the way to measure the error is not very relevant. This is why this method is more suited to correcting a model bias rather than learning from scratch a constitutive model.

That being said, it means that two solutions must be implemented. First, a relevant initialization should be performed before minimizing the mCRE. It is quite natural then to use transfer learning, with a first supervised initialization training to represent an a priori constitutive model assumed to be close to the real behavior. Second, the optimization should be constrained in order not to go too far from the initialization in one step. This is done by constraining the value of the learning rate to do only small steps (see the tuning of the learning rate in the previous chapter).

These two tricks are only solving a part of the problem. Indeed, constraining the optimization process to perform only small steps prevents the exploration of different mCRE valleys. Thus, without being careful, a bias can be introduced with the initialization training, because the optimization process might finish in a local minima. The following section illustrates this phenomenon, referred to as "initialization bias".

1.2 An example of initialization bias

This section aims to illustrate the problem of initialization bias with an example.

Problem definition

The problem treated is similar to the first problem treated in Chapter 2, Section 5.1.

Synthetic data are generated following the following model:

$$\psi(\epsilon) = \frac{1}{2}E_1^+ \langle \epsilon_{11} \rangle_+^2 + \frac{1}{2}E_1^- \langle \epsilon_{11} \rangle_-^2 + \frac{1}{2}E_2 \epsilon_{22}^2 + G \epsilon_{21}^2 \quad (4.1)$$

$\langle \bullet \rangle_+$ and $\langle \bullet \rangle_-$ stand respectively for the positive and negative parts, $E_1^+ = 12GPa$ and $E_1^- = E_2 = 20GPa$ are the tension and compression Young moduli, $G = 8GPa$ is the Coulomb modulus and ϵ_{ij} are components of the linearized strain tensor ϵ .

Let us say that the model used for prediction is the following

$$\psi(\epsilon) = \psi_{NN}(\epsilon_{11}) + \frac{1}{2}E_2 \epsilon_{22}^2 + G \epsilon_{21}^2 \quad (4.2)$$

with $E_2 = 20GPa$ the Young modulus in direction 2, $G = 8GPa$ the Coulomb modulus. The parameters E_2 and G are assumed to be known (not updated), and the goal of this example is to find the parameters of $\psi_{NN}(\epsilon_{11})$ which is a function described by an input-convex neural network. This decomposition, which might seem surprising, is only introduced for the sake of simplicity to illustrate the initialization bias phenomenon.

Now let us say that ψ_{NN} is initialized with the following model:

$$\psi_{AP} = \frac{1}{2}E_1 \epsilon_{11} \quad (4.3)$$

with $E_1 = 20GPa$.

Manual initialization in two different ways.

In the following, in order to illustrate the point, the initialization of ψ_{NN} will be made manually (without supervised training), in two different ways. Later the mCRE training is performed with these two initializations to show the strong influence of the initialization bias. Figure 4.2 illustrates the two different initialization strategies.

The two different strategies, presented in Figure 4.2, give the same function at the initialization, which is ψ_{AP} .

mCRE minimization for the two different initialization

Now that the network is initialized with an a priori model (thus replacing the supervised initialization training in this example), the mCRE minimization can be performed. In this example, no evolution law is prescribed so the method of Chapter 2 is implemented. Let us recall that the learning rate is adapted with the empirical rules detailed in Section 3.3 of Chapter 2. In the implemented example, the database is composed of one loading step, for the same bending beam as the one in Figure 2.4, but with full-field displacement measurements. Figure 4.2 shows the results of the mCRE minimization for the two different initializations. Even though the two initializations represent the same a priori function, only one of them (initialization (b)) converges to the true potential. For the initialization (a),

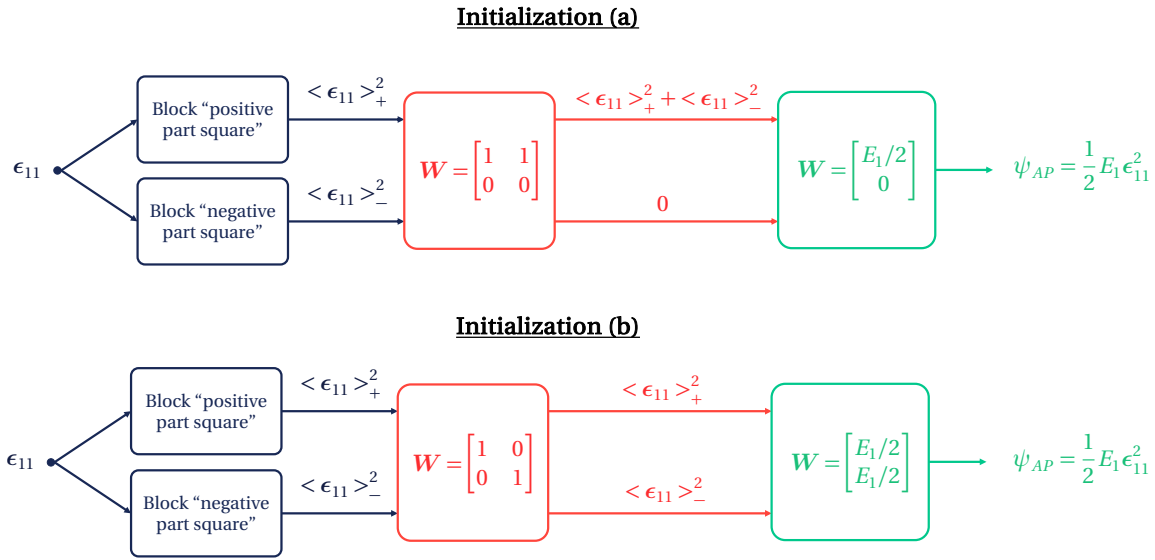


Figure 4.1 • Two different possibilities to initialize ψ_{NN} to represent ψ_{AP}

it can be observed that the Young modulus is the same for the tension and the compression, as the parabola is symmetric. When looking at the structure of the initialized weights, it makes sense that initialization (a) cannot separate tension from compression, whereas it is possible with initialization (b). Because the learning rate is constrained to make only small gradient steps, the optimization failed to switch minimization valleys. Therefore, a good representation of the a priori model does not ensure the convergence of the minimization to the global minimum. This example might seem trivial, but when using supervised a priori training instead of manual initialization, it seems very complicated to predict whether the optimization will reach a global or a local minimum. To conclude, this example has illustrated the phenomenon of initialization bias: two initializations representing the same a priori model can give two different convergences of the mCRE minimization. The next section discusses the possibilities to overcome this limitation.

1.3 Possibilities to get rid of initialization bias

There are several ways of avoiding this initialization bias. The first idea is to carry out several mCRE trainings with different initializations. This idea can be found in the NN-ECULID method (Thakolkaran et al., 2022), in which 30 training runs are performed with 30 different initializations. Once the 30 training sessions have been completed, only the best model is retained. Although functional, this technique is not at all optimized due to its high computational cost.

Another idea is to initialize in the right optimization valley, *i.e.* the one in which the minimum is the global minimum. This idea seems very complicated in practice, because the cartography of the loss is not available. Furthermore, one can say that if we knew where is the global minimum, we would probably not need neural networks to learn a constitutive model.

Finally, a third idea is to work on the parameterization of the network (number of layers

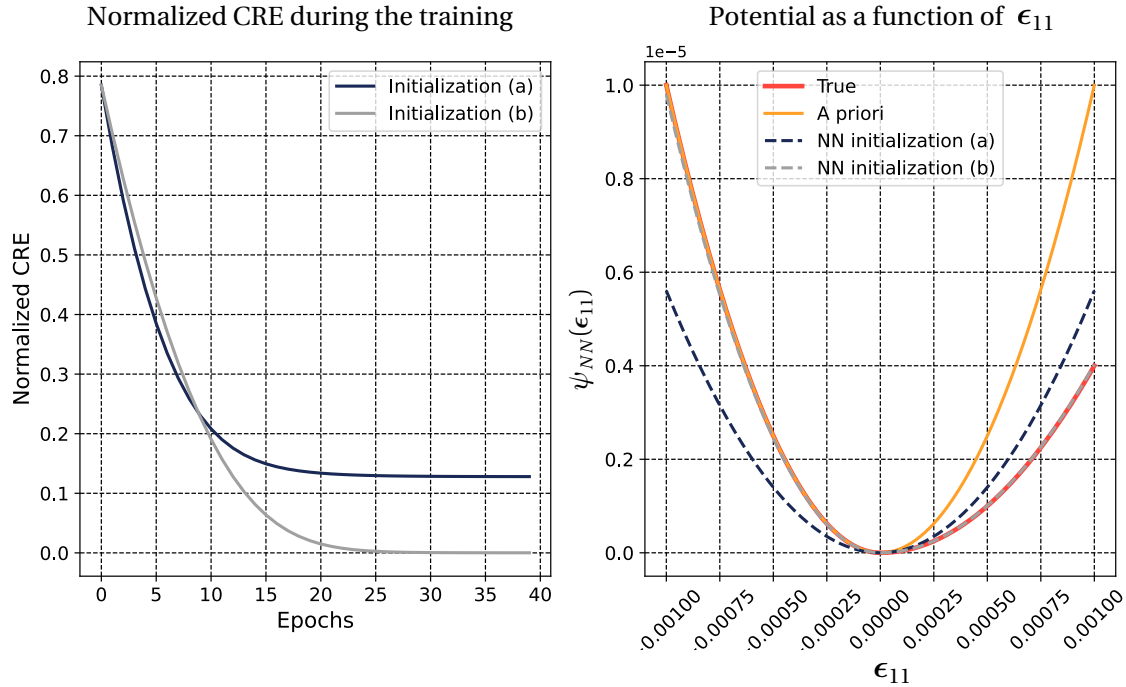


Figure 4.2 • Left: Evolution of the normalized CRE during the mCRE minimization. Right: ψ_{NN} as a function of ϵ_{11} before and after training for different initialization.

and number of neurons by layers) to try to avoid this initialization bias. In the example presented in Section 1.2, the problem came from the fact that the network had multiple layers. The idea of the next section is to use a one-layer input-convex neural network.

2 An adequate parameterization of neural networks for constitutive model

The previous section highlighted that multiple-layer neural networks can create an initialization bias with the minimization of the mCRE. This initialization sensitivity is emphasized by two constraints of the mCRE minimization: the need for a meaningful first model (required to compute admissible fields since the first iteration), and the need for a small learning (required not to go too far too quickly from the meaningful initialization). As this problem is inherent to the mCRE minimization, it makes sense to explore a solution that is not classical in Deep Learning. The proposed solution is to investigate a one-layer modified input-convex neural network in order to get rid of the initialization bias. This idea might seem to be a counter-current philosophy as the trend in Deep Learning is to constantly increase the network depth. Indeed, it is known that increasing the depth of the network increases the expressivity, which is the ability to learn complex functions. Nevertheless, in the Generalized Standard Material framework, the functions to learn are smooth because they are convex. Thus, a strong expressivity is probably not needed, and a one-layer modified ICNN may be sufficient. After introducing the concept of one-layer modified ICNN

in Section 2.2, Section 2.2 investigates whether a one-layer modified ICNN is expressive enough or not. A common problem with neural networks is their lack of interpretability, because of their important number of layers. In the case of one-layer modified ICNN, we show that a certain amount of interpretability can be reached, especially regarding extrapolation: Section 2.3 discusses the extrapolation.

2.1 A one-layer modified ICNN

The idea of this work is to propose a simple parameterization to get rid of initialization bias and improve interpretability. Before introducing this parameterization, let us briefly recall the requirements that should be satisfied by the function ψ_{NN} described by a neural network.

- ψ_{NN} should be convex with respect to the input;
- The second-order derivative of the network with respect to the input should not be zero everywhere (because Step 1 in the mCRE minimization requires the computation of tangent operators).

This parametrization slightly modifies the one-layer ICNN as it was originally introduced. The output of one-layer ICNN as proposed in (Amos et al., 2017) writes:

$$\psi(\mathbf{x}) = \sum_{i=1}^N W_2^i a(\mathbf{W}_1^i \mathbf{x}^T + b^i) \quad (4.4)$$

with $\mathbf{x} \in \mathbb{R}^d$ the neural network input, $\psi \in \mathbb{R}$ the neural network output, $N \in \mathbb{N}$ the number of neurons in the hidden layer, $\mathbf{W}_1^i \in \mathbb{R}^d$ the i -th line of the first layer weights matrix \mathbf{W}_1 , $b_1^i \in \mathbb{R}$ the i -th component of the first layer bias vector \mathbf{b}_1 , $W_2^i \in (\mathbb{R}^+)^d$ the i -th component of the second layer weights vector \mathbf{W}_2 and a a component-wise activation function. The positivity constraints on \mathbf{W}_2 and the use of a convex and non-decreasing activation function ensure the convexity of ψ with respect to the input. In (4.4), the skip connections are not considered as our numerical experiments did not show any benefits from them. In Amos et al., 2017, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1$ are trainable parameters.

The present work relies on the following slightly modified parameterization:

$$\psi(\mathbf{x}) = \sum_{i=1}^N a(\mathbf{W}^i (\mathbf{x} - \mathbf{b}^i)^T) \quad (4.5)$$

with $\mathbf{W}^i \in \mathbb{R}^d$ the i -th line of the weights matrix \mathbf{W} and, $\mathbf{b}^i \in \mathbb{R}^d$ the i -th line of biases matrix \mathbf{b} . In this parametrization, the values of \mathbf{b} are fixed with a physical argument, and only the matrix \mathbf{W} is trainable. The following paragraph details how to fix the bias.

Sampling of the biases with physical knowledge

A constitutive model should always be defined with a domain of validity. For example, when defining an elastic model, it is advisable to define a domain in which the material is elastic. This must also be checked when the constitutive model is described by a neural network. For this reason, we have chosen to fix the bias values a priori and not to make them trainable. In the following, the biases are sampled with the following procedure:

1. A range of validity is defined a priori;
2. A number of neurons in the layer is chosen by the user;
3. The biases are uniformly sampled in this validity range.

Use of Softplus activation function to overcome the zero second order derivative issue

The architecture proposed in (4.5) has a second-order derivative of the output with respect to the input that is zero over the entire definition domain. As a result, this model violates the requirements mentioned at the beginning of this section. To overcome this problem, the ReLU activation function can be replaced by a Softplus activation function (which is a regularized version of the ReLU with a non-zero second-order derivative):

$$\text{Softplus}(x) = \frac{1}{\beta} \log(1 + \exp(\beta x)) \quad (4.6)$$

where β is a parameter acting on the regularization level of the ReLU as illustrated in Figure 4.3.

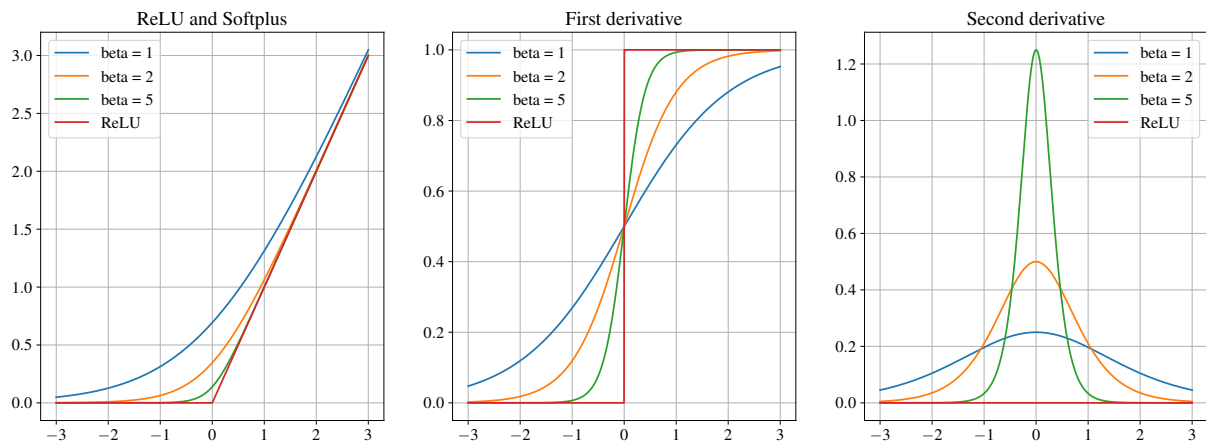


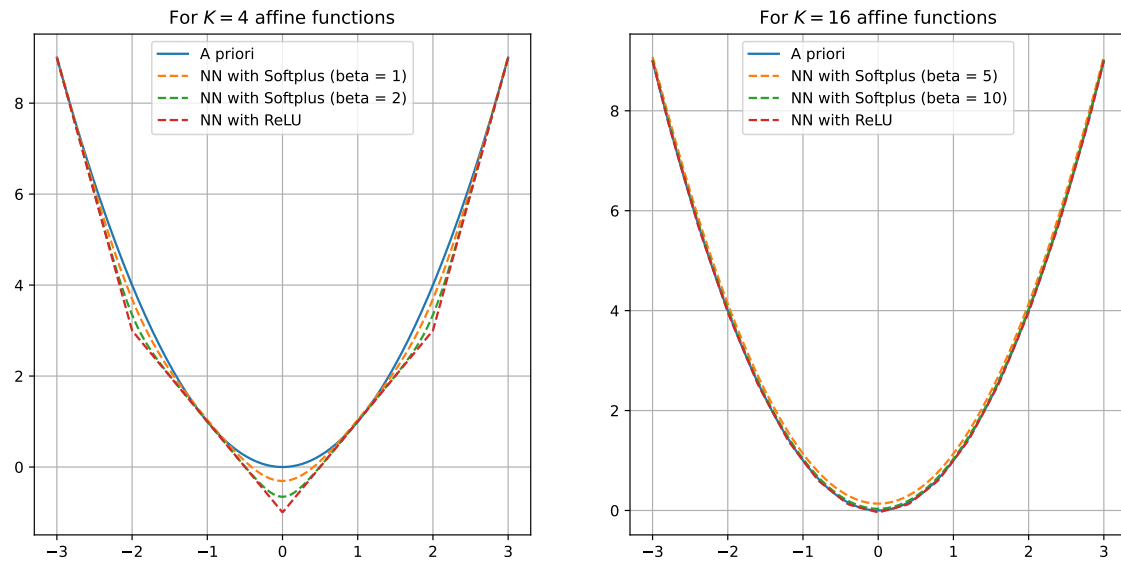
Figure 4.3 • Comparison of the ReLU and Softplus function for several values of β . Left: Softplus and ReLU functions. Middle: First derivative. Right: Second-order derivative.

📖 Remark

In Chapter 2, the activation function was chosen as ReLU square ($a(X) = \langle X \rangle_+^2$), and in Chapter 3 (which was developed later), the Softplus function was used. This change

is explained by the fact that ReLU square, coupled with the positivity constraints needed by the ICNN (Amos et al., 2017), does not allow for learning every convex function (one example being linear functions.)

Example: This example shows how the function $\psi_{AP}(x) = x^2$ can be approximated by an ICNN with ReLU and Softplus (for multiple values of β) activation functions, using the parameterization described so far. With the use of a Softplus activation function, the represented function is no longer piecewise linear.



The value of β can be chosen following the knowledge of the discretization. A possibility is to perform a supervised initialization training to represent an a priori model. As the supervised training does not require the use of non-vanishing second-order derivatives, it can be performed with ReLU activation functions. At the end of this supervised training, it is possible to change the ReLU to Softplus and to choose β in the following way. It is worth noting that β acts as a "response time" in the exponential. Let us denote Δx the discretization step defined by:

$$\Delta x = \frac{\text{validity range}}{\text{number of neurons}}$$

Looking at the previous example, if we want the output of the activation function to be equal to the Softplus function only on its own interval and equal to ReLU elsewhere, we can write the following system:

$$\left\{ \begin{array}{l} \left. \frac{\partial \text{Softplus}(x)}{\partial x} \right|_{-\frac{\Delta x}{2}} < \epsilon \\ \left. \frac{\partial \text{Softplus}(x)}{\partial x} \right|_{\frac{\Delta x}{2}} > 1 - \epsilon \end{array} \right\} \quad (4.7)$$

where ϵ is a user-defined tolerance. Solving this system gives an upper bound to choose

β . A lower bound can be obtained in the same way writing a minimal condition on the second-order derivative at the middle of the interval.

2.2 Expressivity of one-layer modified ICNN

This section aims to check if the proposed parameterization is expressive enough. To do so, the proposed architecture is compared with multiple-layer ICNN, roughly composed of the same number of parameters. Table 4.1 details the number of parameters for the 4 architectures tested.

| Model type | Number of parameters |
|---------------------------|----------------------|
| Proposed parameterization | 4096 |
| ICNN with 2 layers | 4548 |
| ICNN with 3 layers | 4516 |
| ICNN with 4 layers | 3732 |

Table 4.1 • Number of parameters for the 4 architectures tested.

To compare the expressivity of these architectures, they have been compared on four different test cases of convex functions illustrated in Figure 4.4.

The four convex functions exhibit various characteristics that differentiate them in terms of smoothness, differentiability, and behavior with respect to directionality. Function f_1 represents a quadratic function that is smooth and differentiable everywhere, as it is a polynomial function. f_2 , representing the Euclidean norm, introduces an additional square root operation, resulting in a lack of differentiability at the origin. f_3 and f_4 are different from the first two as they exhibit different directional behaviors. For f_3 the directional behaviors only differ for the director coefficient, while for f_4 it is the power and the director coefficient that are changing.

For each function, each architecture has been trained 7 times with a learning rate ranging from 10^{-7} to 1, on 2000 epochs, with 2^{16} data points, with a batch size of 1024. For each function and for each architecture, Table 4.2 gives the final value of the loss function (mean square error), on a validation dataset, for the best of the 7 trainings. Table 4.2 shows that the expressivity of the proposed parameterization is the same order of magnitude as the other architectures with more layers (but with roughly the same number of parameters). This result is explained by the fact that convex functions are smooth functions, so they do not require a very expressive networks to be represented.

2.3 Interpretable extrapolation offered by one-layer modified ICNN

One limitation preventing neural networks from being used in critical applications is their uncertain extrapolation. This difficulty is explained by the complex and often opaque nature of deep neural networks which can make them challenging to diagnose and correct extrapolation errors. The interpretable nature of the extrapolation of a physical model is a strong advantage over the use of deep neural networks. Nevertheless, with the proposed

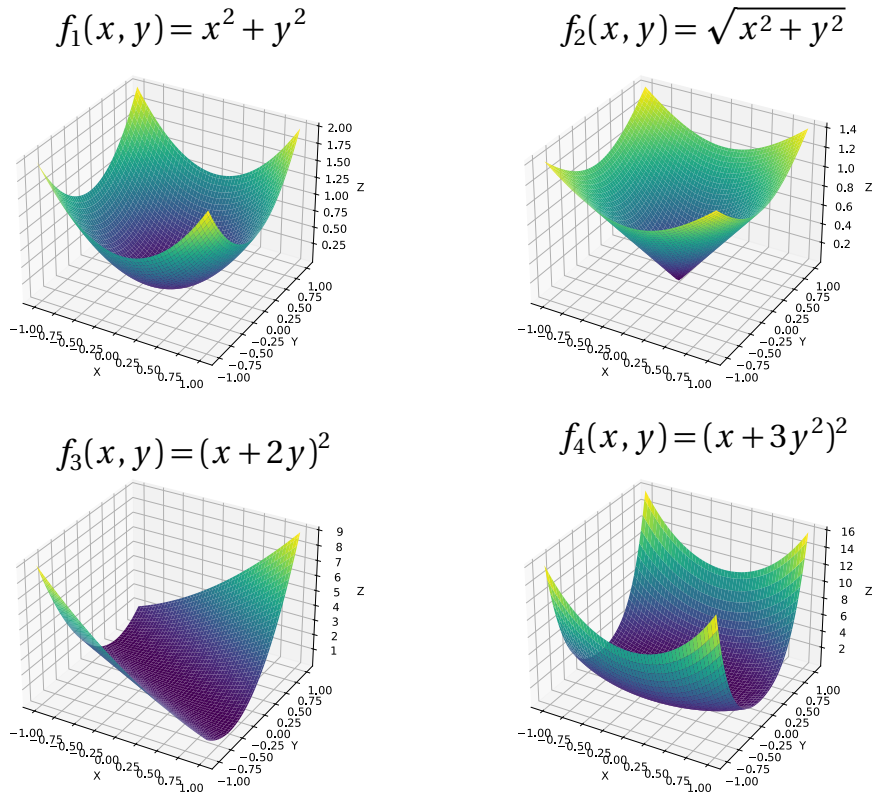


Figure 4.4 • 4 convex functions used to assess the expressivity of the 4 architectures compared.

| Model type | Function 1 | Function 2 | Function 3 | Function 4 |
|---------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Proposed parameterization | 1.92×10^{-7} | 1.06×10^{-6} | 3.67×10^{-7} | 8.54×10^{-6} |
| ICNN with 2 layers | 2.81×10^{-6} | 2.26×10^{-7} | 6.79×10^{-6} | 8.31×10^{-5} |
| ICNN with 3 layers | 1.62×10^{-6} | 2.19×10^{-7} | 3.36×10^{-6} | 8.98×10^{-5} |
| ICNN with 4 layers | 1.25×10^{-6} | 1.20×10^{-6} | 4.25×10^{-5} | 3.36×10^{-4} |

Table 4.2 • Value of the validation loss function at the end of the best training for each of the 5 architectures tested.

parameterization relying on a one-layer network, the extrapolation can be interpreted. This section aims to discuss the extrapolation offered by this parameterization.

To discuss the extrapolation, *i.e.* the prediction for input not available in the training database, let us have a look at the gradient of the mCRE with respect to the parameters (in the case of a nonlinear elastic behavior):

$$\frac{d\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \mathbf{p})}{d\mathbf{p}} = \int_{\Omega} \left(\frac{\partial \psi}{\partial \mathbf{p}} \Big|_{E(\hat{\mathbf{u}})} - \frac{\partial \psi}{\partial \mathbf{p}} \Big|_{E(\hat{\mathbf{v}})} \right) \quad (4.8)$$

The field $\hat{\mathbf{u}}$ is constructed to be close to the observations up to the noise level and $\hat{\mathbf{v}}$ is an admissible field obtained with the finite element solution for the constitutive model parameterized by \mathbf{p} .

Let us consider one parameter \mathbf{W}^k and its associated weight \mathbf{b}^k .

$$\left. \frac{\partial \psi}{\partial \mathbf{W}^k} \right|_E = \mathbf{W}^k a'(\mathbf{W}^k(\mathbf{E} - \mathbf{b}^k)^T) \quad (4.9)$$

$$(4.10)$$

where,

$$a'(\mathbf{W}^k(\mathbf{E} - \mathbf{b}^k)^T) = \begin{cases} \approx 0 & \text{if } \mathbf{W}^k(\mathbf{E} - \mathbf{b}^k)^T \ll 0 \\ \approx 1 & \text{if } \mathbf{W}^k(\mathbf{E} - \mathbf{b}^k)^T \gg 0 \end{cases} \quad (4.11)$$

For a weight for which its associated bias is far from the training database, the gradient will always be zero, which means that the weight will never be updated. This conclusion is very interesting because it means that the weights far from the training database are the same as the one at the end of the supervised initialization training to represent an a priori model. Far from the observations, the extrapolation is therefore performed with the a priori model. The model is only corrected where there are observations. This property is a strong advantage of the proposed parameterization because one can argue that it is better to extrapolate with a wrong but understood physical model than with a non-understood neural network.

2.4 Discussion on the use of second-order optimization method

In Deep Learning, most of the trainings are performed with gradient descent (or variants such as Adam). Second-order methods, like Newton's method (or variants such as quasi-Newton, BFGS) are rarely used. For a gradient-based optimization, weights are updated with $\mathbf{p}_{k+1} = \mathbf{p}_k - l_r \nabla_{\mathbf{p}} \mathcal{L}$. Newton's method consists of writing a second-order Taylor development around the set of parameters \mathbf{p}_k :

$$T(\mathbf{p}) = \mathcal{L}(\mathbf{p}_k) + \nabla_{\mathbf{p}} \mathcal{L}(\mathbf{p}_k)(\mathbf{p} - \mathbf{p}_k) + \left. \frac{\partial^2 \mathcal{L}}{\partial \mathbf{p}^2} \right|_{\mathbf{p}_k} (\mathbf{p} - \mathbf{p}_k)^2 \quad (4.12)$$

The parameters \mathbf{p}_{k+1} are then obtained by looking for the stationarity of the Taylor development, *i.e.* the parameters where the gradient of the Taylor development is zero:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \left(\left. \frac{\partial^2 \mathcal{L}}{\partial \mathbf{p}^2} \right|_{\mathbf{p}_k} \right)^{-1} \nabla_{\mathbf{p}} \mathcal{L}(\mathbf{p}_k) \quad (4.13)$$

One advantage of Newton's method over gradient-descent is that they do not require the choice of a learning rate, which would allow to get rid of the empirical rules used in this manuscript. This method is not common in Deep Learning for multiple reasons. First, Newton's methods are efficient for strictly convex loss function, which is very rarely the

case in Deep Learning (Lim & Lim, 2022). Secondly, it requires the computation of the Hessian which is a square matrix of if size number of parameters. For neural networks with an important number of parameters, the computation of the Hessian is already expensive (methods such as BFGS alleviate this concern by computing an approximation of the Hessian). Then the Hessian needs to be inverted, which is even more computationally expensive. Finally, gradient-descent methods are more suited for GPU computation as they do not need a matrix inversion.

In the present case, because the network is not deep with the proposed parameterization, the choice of gradient descent might be questioned. Indeed, as mentioned in Chapter 1, the mCRE is known for its convexity properties. With the proposed parameterization the number of parameters is reasonable and the Hessian is sparse (see the second-order derivative of the activation function in Figure 4.3). With the mCRE the Hessian is computed by derivating one time the gradient computed with the admissible fields of Step 1. An implementation of Newton's method to minimize the mCRE has been performed to compare with gradient-descent. The results were not convincing compared to gradient descent. Even if the proposed parameterization offers a better convexity of the loss function, the loss function is not strictly convex everywhere which explains the poor results of Newton's method.

3 Results with one-layer modified ICNN

In Section 2, two main advantages of the parameterization have been discussed: the possibility of getting rid of initialization bias and the possibility of extrapolating with the a priori knowledge. In the present section, these two advantages are checked on numerical examples. The first example is the same as the one used to illustrate the initialization bias in Section 1.2 and the second one is an example in elastoplasticity.

3.1 Example with a nonlinear elastic behavior

Section 1.2 highlighted the possible presence of initialization bias with multiple layer input-convex neural networks. This section aims to show that the proposed network parameterization does not suffer from this problem anymore.

The problem treated was the following. Synthetic data is generated following the following model:

$$\psi(\epsilon) = \frac{1}{2} E_1^+ \langle \epsilon_{11} \rangle_+^2 + \frac{1}{2} E_1^- \langle \epsilon_{11} \rangle_-^2 + \frac{1}{2} E_2 \epsilon_{22}^2 + G \epsilon_{21}^2 \quad (4.14)$$

and the model used for prediction is the following

$$\psi(\epsilon) = \psi_{NN}(\epsilon_{11}) + \frac{1}{2} E_2 \epsilon_{22}^2 + G \epsilon_{21}^2 \quad (4.15)$$

The goal of this example is to find the parameters of $\psi_{NN}(\epsilon_{11})$ which is a function described

by an input-convex neural network initialized with:

$$\psi_{AP} = \frac{1}{2} E_1 \epsilon_{11}^2 \quad (4.16)$$

In the following, ψ_{NN} is a one-layer modified ICNN with 10 neurons. The biases are uniformly sampled in $[-2 \times 10^{-3}; 2 \times 10^{-3}]$.

Figure 4.5 shows the function ψ_{NN} at initialization (as well as ψ_{AP} and the true potential), and the normalized CRE in the bending beam at the beginning of the mCRE minimization. The model bias is localized in the part of the beam that is in tension, which is explained by the fact that the potential described by ψ_{NN} at the initialization is different from the true one for the positive values of ϵ_{11} .

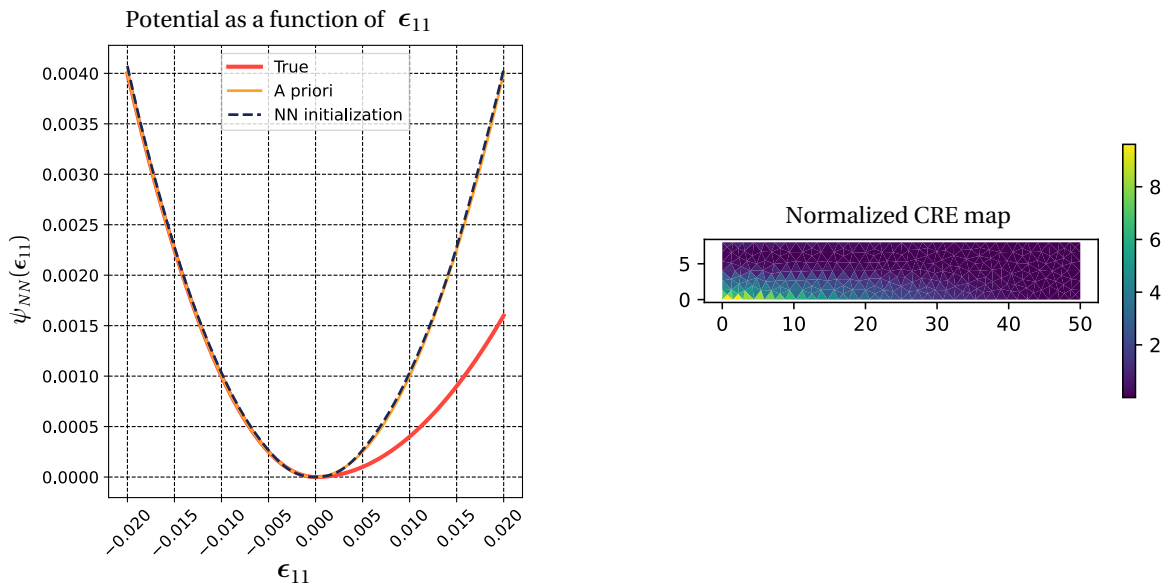


Figure 4.5 • Left: ψ_{AP} , the true potential, and ψ_{NN} at initialization. Right: normalized CRE map at initialization.

The training is performed according to the procedure described in Chapter 2, with the parameterization proposed in the present chapter. This training has been replayed 10 times with different initialization training to represent the same ψ_{AP} and the results at convergence are similar (same values of CRE and mCRE, same learned potential). Figure 4.6 summarizes the mCRE minimization, for one of these 10 training, with convergence curves of the normalized CRE, the normalized data loss and the mCRE, as well as the potential at the end of the training, and the map of the normalized CRE at the end of the mCRE minimization. The result of one of these trainings shows that the initialization bias described in Section 1.2 is no longer present with the proposed parameterization.

3.2 Example of interpretable extrapolation in elastoplasticity

In Section 2.3, it was shown that the proposed parameterization offers the possibility to perform an interpretable extrapolation. Indeed, with the one-layer modified ICNN (fixed bias

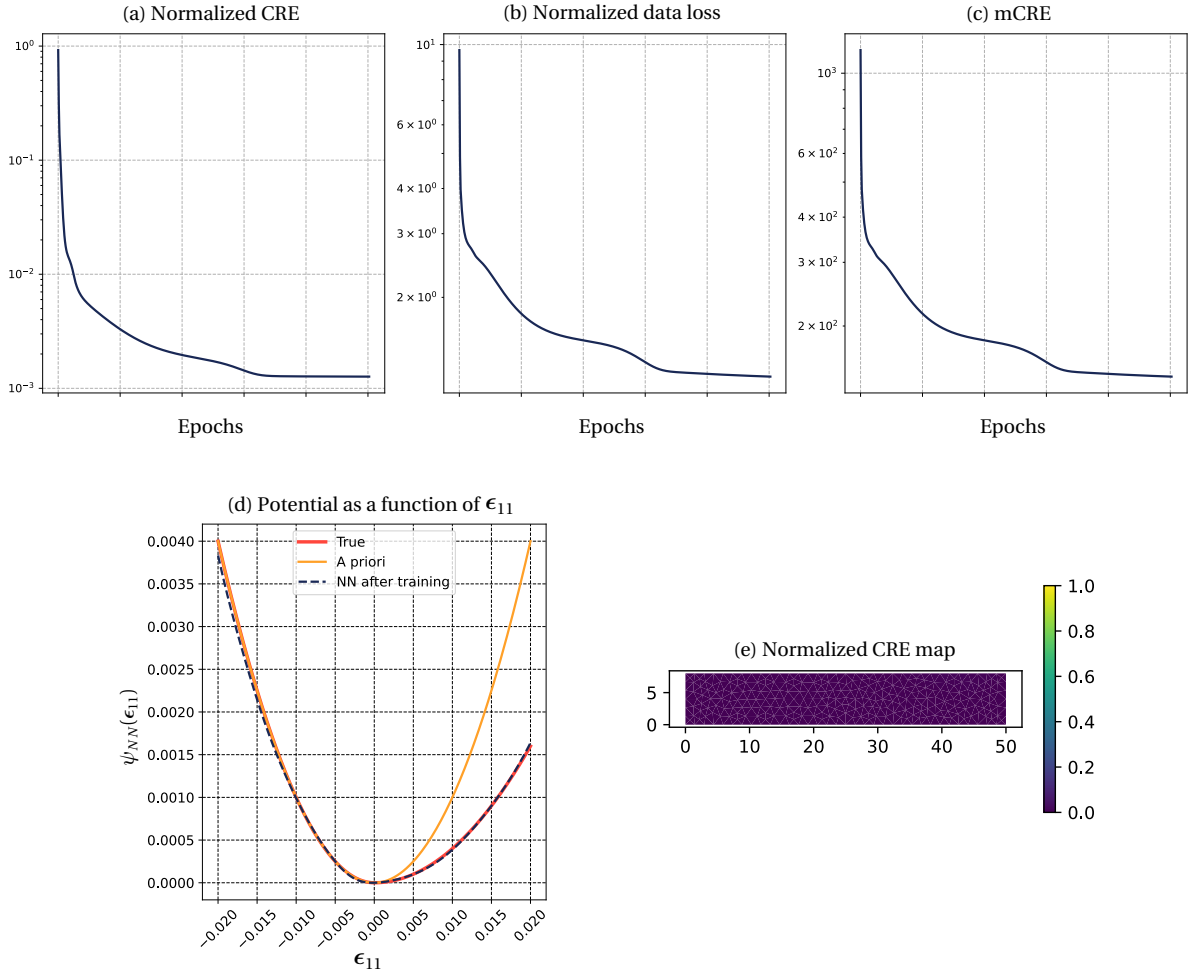


Figure 4.6 • (a) Evolution of the normalized CRE. (b) Evolution of the normalized data loss. (c) Evolution of the mCRE. (d) ψ_{AP} , the true potential, and ψ_{NN} at the end of the mCRE minimization. (e) Normalized CRE map at the end of the mCRE minimization.

and Softplus activation function), the weights associated with biases far from the training database input are not updated. As the network is initialized with an a priori model before the mCRE minimization, it means that once the mCRE is minimized, the extrapolation far from the training database is performed with the a priori model.

This section shows the extrapolation of the elastoplastic example of Chapter 3 Section 3.2 (learning of nonlinear isotropic hardening law). In this example, only the plastic part of the free energy is learned. The network is initialized with $\psi_{p,AP}(p) = \frac{1}{2}hp^2$ and the training data was generated with $\psi_p(p) = Ap + \frac{A}{B}(e^{-Bp} - 1)$. Before the mCRE minimization, the biases are sampled in $[0; 8 \times 10^{-4}]$, and a supervised training is performed with database of couples $\{p, \psi_{p,AP}\}$, where p is also uniformly sampled in $[0; 8 \times 10^{-4}]$.

Figure 4.7 shows that the initialized neural network correctly represents ψ_{AP} for $p \in [0; 8 \times 10^{-4}]$, as well as the hardening law (R as a function of p). When the beam of Section 3.2 (Chapter 3), is loaded in tension, the behavior σ_{11} as a function of ϵ_{11} is also properly

represented.

After this initialization training, the mCRE minimization is performed, but this time with a loading such that $p \in [0; 2.5 \times 10^{-4}]$. Figure 4.7 shows that:

- For $p \in [0; 2.5 \times 10^{-4}]$, the network after training (mCRE minimization) properly represents the nonlinear hardening law;
- For $p \in [2.5 \times 10^{-4}; 8 \times 10^{-4}]$, the hardening law is linear, which corresponds to the a priori model.

A similar conclusion can be observed in σ_{11} as a function of ϵ_{11} . To conclude, this simple example has illustrated that the extrapolation outside the training database is performed with the a priori model. Nevertheless, this requires carefully selecting the interval in which the biases are sampled to make sure this interval is wider than what will be seen during the mCRE minimization.

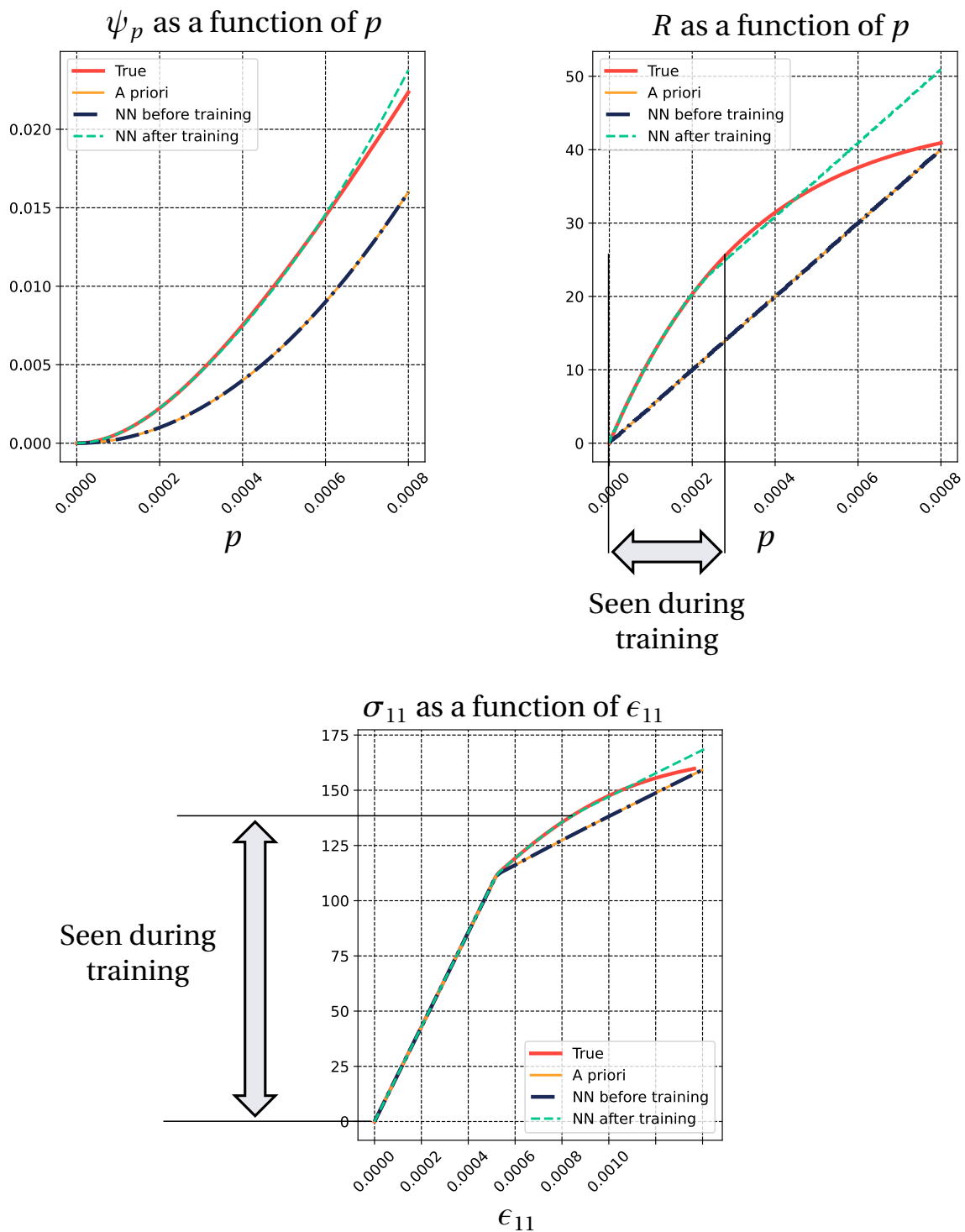


Figure 4.7 • Top left: ψ_p as a function of p . Top right: R as a function of p . Bottom σ_{11} as a function of ϵ_{11} . For each plot, the true and a priori behavior are represented, as well as the neural network before and after the mCRE minimization.

Chapter conclusion

This chapter introduced an adequate parameterization suited for the mCRE minimization. This parameterization was motivated by the need to get rid of the initialization bias. A close initialization is crucial with the mCRE in order to compute meaningful admissible fields at the first epoch. An example has demonstrated that the initialization bias can be particularly problematic with the mCRE minimization because it requires the optimization to be constrained in order not to go too far from the initialization in one step.

The chosen network was based on a one-layer modified ICNN, with a priori chosen and fixed biases, Softplus activation function (with the β parameter fixed with the knowledge of discretization) to avoid vanishing second-order derivatives. The expressivity of this architecture was compared to deeper ICNN. This comparison showed that this architecture is expressive enough to represent convex functions. Finally, significant advantages of the parameterization were discussed and illustrated on examples. This architecture allows getting rid of the initialization bias, as well as the possibility to extrapolate with the a priori model.

5

A Modified Dual Kalman Filtering approach for sequential data-assimilation involving nonlinear evolution laws

The development of this chapter has been done in collaboration with Sahar Farahbakhsh (involved in another ERC project thesis). The work presented in this chapter has been the subject of the following contribution:

Benady A, Farahbakhsh S, Baranger E, Poncelet M, Chamoin L. A Modified Dual Kalman Filtering approach for sequential data-assimilation involving nonlinear constitutive evolution laws. (in preparation). ¹

Contents

| | | |
|----------|--|------------|
| 1 | From Linear Kalman Filter to Modified Dual Kalman Filter | 112 |
| 1.1 | Linear Kalman Filter (LKF) | 113 |
| 1.2 | Nonlinear extensions of KF: focus on the Scaled Spherical Simplex Filter (S3F) | 114 |
| 1.3 | Kalman Filter for parameter identification | 117 |
| 2 | Extension of MDKF to address evolution laws | 118 |
| 3 | Results | 120 |
| 4 | Toward a coupling between MDKF and model bias correction with neural networks | 120 |

¹Sahar Farahbakhsh developed during her thesis a numerical method based on the Modified Dual Kalman Filtering, to sequentially update model parameters of linear constitutive model (without evolution laws). This chapter aims to merge her work with mine (on the learning of evolution laws) to sequentially update the parameters of nonlinear evolution laws.

The previous theses on the mCRE (Diaz, 2023; Marchand, 2017) have shown that a strong advantage of the mCRE lies in the possibility to treat both the model updating and sequential data assimilation in a unified framework. So far, this thesis has extended the mCRE framework for the model updating of history-dependent dissipative constitutive laws. The goal of this chapter is to show that the development of this thesis can still be coupled with a sequential data assimilation procedure, thus preserving the possibility of relying on a unified framework. Therefore the development of this chapter is only a preliminary work to illustrate the possibility of integrating the evolution laws in the sequential data assimilation procedure.

The continuous prediction of structural state requires a sequential data assimilation framework to interface the numerical model with the on-the-fly observations. First introduced and manually applied in (Richardson & Lynch, 1922) by a meteorologist, data assimilation combines physical laws, or models, and observations to obtain the best possible estimation of the system state. Data assimilation methods were commonly used at first in oceanography and meteorology sciences for weather prediction, but they have been applied in other fields of science such as engineering throughout the years. Sequential data assimilation methods are probabilistic approaches that provide estimates of the entire system state as well as its associated uncertainties by propagating information forward in time (Bertino et al., 2003). The Best Linear Unbiased Estimator (BLUE) is a well-known sequential data assimilation method (Aitken, 1935). Other examples include Bayesian filtering approaches (Law et al., 2015), but in these cases, the calculation of the likelihood functions can be challenging or computationally expensive. Originating from Bayesian inference, the Kalman Filter (KF) addresses this issue by assuming independent Gaussian distributions for every parameter (Kalman, 1960), and stands out as a well-established data assimilation algorithm incorporating a prediction-correction scheme. It iteratively refines predictions using acquired measurements at each step.

Over the years, several extensions to the original linear Kalman Filter have been proposed to enhance its performance across diverse and especially nonlinear applications. The Extended Kalman Filter (EKF) is a very popular extension of Kalman Filters for capturing nonlinearities (Sunahara & Yamashita, 1970; Bucy & Senne, 1971). This method is based on operator linearization by means of Taylor expansions. Other KF extensions mostly rely on the propagation of sampling points to represent the nonlinearities of the system. Some of the common examples are Ensemble Kalman Filters (EnKF) (Evensen, 1994, 2003) and Particle Filters (Del Moral, 1996) where at each time step a sampling made by the Monte Carlo method is propagated through nonlinear system operators, although these methods can become costly due to the sampling approach. Motivated by the need for computational efficiency, the Unscented Kalman Filter (Julier & Uhlmann, 1997; Julier et al., 2000) was developed based on the Unscented Transform (Julier et al., 1995), resulting in a reduced size in sampling point sets. Recently, an even more computationally efficient approach named Scaled Spherical Simplex Filter (S3F) was introduced (Amir et al., 2022; Papakonstantinou et al., 2022) which further reduces the size of the required sample point set in comparison

to the UKF, providing the same level of accuracy.

Although primarily recognized as a state prediction algorithm, certain extensions of the Kalman Filter have found utility as parameter identification algorithms and for solving inverse problems. For this mean, parameters to be identified can be concatenated with state variables in the KF framework, resulting in a Joint Kalman filter. Another possibility is to keep the parameters in the state vector; and turn the observation operator into a state evolution operator based on a second Kalman filter. This approach is called the Dual Kalman filter. Some examples of research using nonlinear extensions of Kalman Filters for identification purposes through joint or dual definitions of the dynamic system can be found in (Mariani & Corigliano, 2005; Gove & Hollinger, 2006; Mariani & Ghisi, 2007; Onat, 2019). It should be noted that despite their proven computational efficiency, Kalman Filters have generally shown sensitivity to noisy measurements for both state estimation and parameter identification, and are susceptible to divergence from the actual solution in the presence of corrupted measurements (Nguyen et al., 2008; Li et al., 2016).

On the contrary, the mCRE framework is robust to noisy and corrupted measurements. Despite its robust nature, the mCRE method is unsuitable for online model updating because it lacks the sequential framework required for such applications. Furthermore, it can be computationally expensive as it requires a two-step and iterative minimization process. To address these limitations, a proposed solution involves integrating the mCRE within a sequential data assimilation method, specifically the Kalman filtering technique. By doing so, the integration within these algorithms overcomes the limitations of both mCRE and Kalman Filters, resulting in a fast and robust sequential model updating method called the Modified Dual Kalman Filter (MDKF). This was proposed in (Marchand et al., 2016) where the MDKF was developed and used for unsteady thermal applications. In (Diaz et al., 2023a) the MDKF was further modified, resulting in a more stable and computationally efficient approach, and verified for low-frequency dynamics cases by applying the developed framework on two earthquake engineering examples. In (Diaz et al., 2023b), the MDKF framework was developed based on UKF and S3F Filters, and the results were compared. The approach using S3F gave more satisfying results from a computational point of view, whereas the accuracy level was the same for both approaches. Figure 5.1 summarizes the classification of Kalman Filter methods.

The present chapter illustrates that the works of (Diaz et al., 2023a) can naturally be extended to integrate evolution laws. In Section 1, the basics of Kalman filtering are recalled, as well as the Scaled Spherical Simplex Filter (S3F) and the extensions of Kalman Filtering to deal with parameter identification (with a focus on MDKF). In Section 2, the MDFK framework is extended to sequentially update model parameters of nonlinear constitutive evolution laws. Section 3 presents the results of this method on an elastoplastic case where an isotropic hardening modulus is updated. Finally, Section 4 opens perspectives to couple neural networks with the MDKF framework.

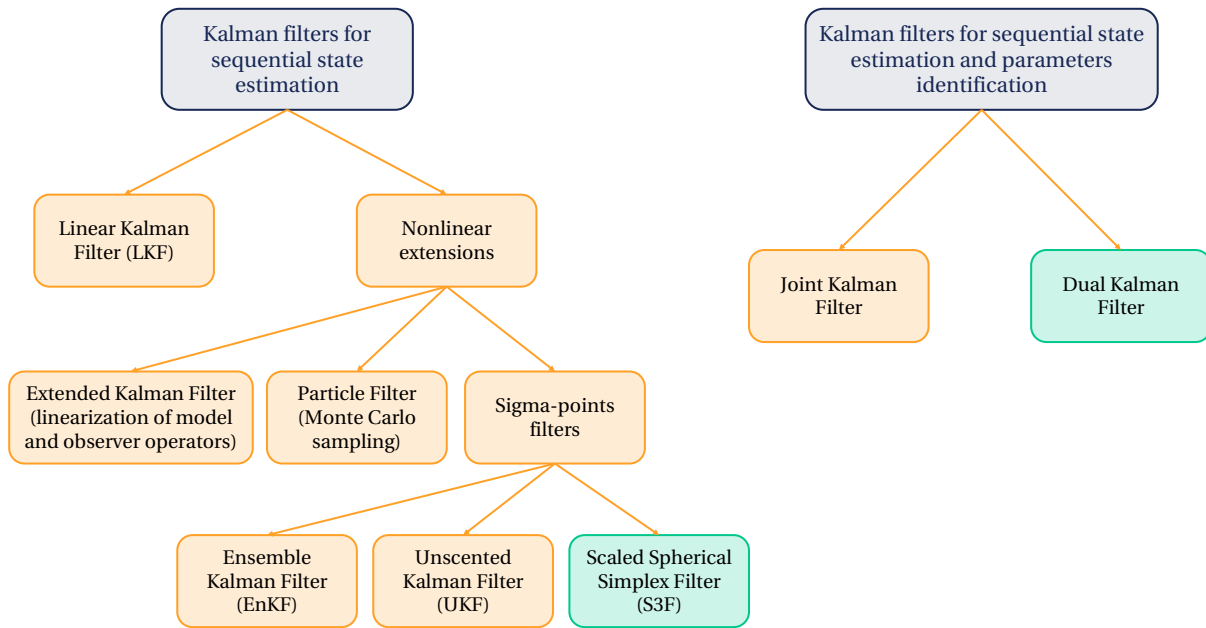


Figure 5.1 • Classification of Kalman Filter methods (not exhaustive). The developments of this chapter use an S3F-based Dual Kalman Filter approach

1 From Linear Kalman Filter to Modified Dual Kalman Filter

This section briefly recalls the concept of Kalman filtering. First of all, let us introduce the notations of the dynamical system under a discrete space-time form required for the Kalman filtering. The system generally consists of two equations, namely a state prediction equation, and an observation equation:

$$\begin{cases} x_k = \mathcal{M}_{k-1}(x_{k-1}, w_{k-1}) \\ z_k = \mathcal{H}_k(x_k, v_k) \end{cases} \quad (5.1)$$

where x_k is the state vector at timestep t_k , z_k the vector of observation data, w_k and v_k are respectively uncorrelated Gaussian errors in model and measurements. Finally, \mathcal{M} and \mathcal{H} are model and observer operators that are derived from the physical problem and are generally nonlinear.

Figure 5.2 schematically shows how Kalman Filters work. Based on this dynamical system, Kalman Filters follow a prediction-correction scheme consisting of two steps. In the prediction step, the system state at the next stage is estimated. This estimation is modified at the second step by new measurements acquired, resulting in a *posteriori* system state.

The remainder of this section is organized as follows. Section 1.1 presents the linear case where both the model and observer operators are assumed to be linear, then Section 1.2 presents the nonlinear KF. Section 1.3 discusses the extension of KF for parameter identification, with an emphasis on the MDKF approach.

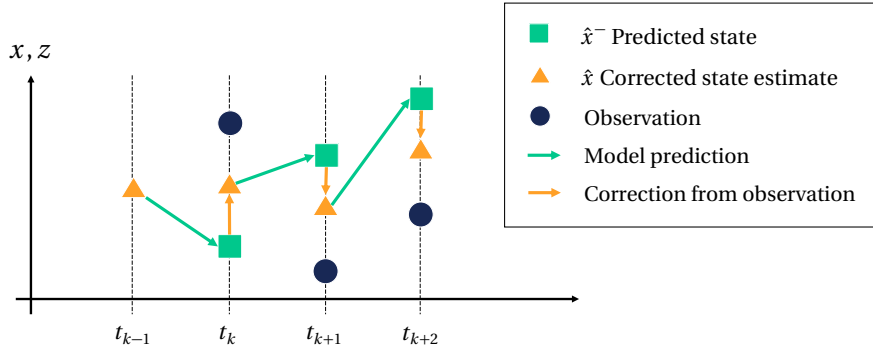


Figure 5.2 • Illustration of the Kalman Filter algorithm

1.1 Linear Kalman Filter (LKF)

The initial Kalman Filter introduced in (Kalman, 1960) considered the model and observer operators to be linear, which turned the equation system (5.1) to (5.2):

$$\begin{cases} x_{k+1} = \mathbf{M} x_k + w_k \\ z_k = \mathbf{H} x_k + v_k \end{cases} \quad (5.2)$$

where \mathbf{M} and \mathbf{H} represent the linear model and observer operators, respectively.

The LKF framework also assumes the modeling and observation error to be statistically independent of each other, and statistically independent of the state. Another assumption is that the model and observation error follow zero-mean normal distributions:

$$\begin{cases} w_k \sim \mathcal{N}(0, \mathbf{Q}) \\ v_k \sim \mathcal{N}(0, \mathbf{R}) \end{cases} \quad (5.3)$$

where \mathbf{Q} is the model error covariance matrix and \mathbf{R} the measurement noise covariance matrix.

Algorithm 3 details the LKF approach. Line 5 corresponds to the model prediction of the next state \hat{x}_k^- based on the estimated current state x_{k-1} and line 6 updates the associated covariance matrix. Line 9 computes the Kalman gain, according to the fact that the best unbiased estimator minimizes the a posteriori error covariance matrix. The term $\mathbf{C}_k^- \mathbf{H}^T$ projects the prior error covariance into the measurement space, providing a measure of how much error we expect in our prediction in the same units as the measurement. The term $\mathbf{H} \mathbf{C}_k^- \mathbf{H}^T$ represents the contribution of the prior state estimate's uncertainty to the measurement's uncertainty, essentially telling how much the uncertainty in the state estimate affects the uncertainty in the measurement output. The term \mathbf{R} is the measurement noise covariance, adding the uncertainty from the measurement itself. The addition of $\mathbf{H} \mathbf{C}_k^- \mathbf{H}^T$ and \mathbf{R} represents the total expected measurement uncertainty. We observe that $\lim_{\mathbf{R} \rightarrow 0} \mathbf{K}_k = \mathbf{H}^{-1}$ and $\lim_{\mathbf{C}_k^- \rightarrow 0} \mathbf{K}_k = 0$ which means that the gain weights the residual more (resp. less) heavily as the measurement error covariance \mathbf{R} approaches zero (resp. as the a priori estimate error covariance \mathbf{C}_k^- approaches zero). In other words, as the measurement

error covariance \mathbf{R} approaches zero, the actual measurement z_k is trusted more and more while the predicted measurement $\mathbf{H} \hat{x}_k^-$ is trusted less and less; in the reverse, as the a priori estimate error covariance \mathbf{C}_k^- approaches zero, the actual measurement z_k is trusted less and less (negligible correction), while the predicted measurement $\mathbf{H} \hat{x}_k^-$ is trusted more and more.

Finally, lines 10 and 11 respectively update the state estimate and the associated covariance matrix.

Algorithm 3 Linear Kalman Filter Algorithm

```

1: Input:
2: Model and observations matrices  $\mathbf{M}$  and  $\mathbf{H}$ , noise covariance matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , initial
   state vector  $\hat{x}_0$  and associated covariance matrix  $\mathbf{C}_0$ 
3: for  $k = 0, 1, 2, \dots$  do
4:   Prediction Step:
5:    $\hat{x}_k^- = \mathbf{M} x_{k-1}$  ▷ Predict next state
6:    $\mathbf{C}_k^- = \mathbf{M} \mathbf{C}_{k-1} \mathbf{M}^T + \mathbf{Q}$  ▷ Predict next covariance
7:
8:   Update Step:
9:    $\mathbf{K}_k = \mathbf{C}_k^- \mathbf{H} (\mathbf{H} \mathbf{C}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$  ▷ Compute Kalman Gain
10:   $x_k = \hat{x}_k^- + \mathbf{K}_k (z_k - \mathbf{C} \hat{x}_k^-)$  ▷ Update estimate with observation
11:   $\mathbf{C}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{C}_k^-$  ▷ Update the estimate covariance
12: end for
    
```

1.2 Nonlinear extensions of KF: focus on the Scaled Spherical Simplex Filter (S3F)

As the LKF does not take into account the possible nonlinearities in the dynamical system, some extensions of the Kalman Filters have been introduced to address this issue. Indeed, the difficulty that arises when considering nonlinear operators is as: given a Gaussian probability density characterized by its mean \bar{X} and its covariance matrix \mathbf{C}_x , how can we construct the a posteriori probability density resulting from the application of a nonlinear operator \mathcal{A} ?

Extended Kalman Filter (EKF): linearization of the model and observer operators

As mentioned in the introduction of this chapter, the Extended Kalman Filter (EKF) is based on a linearization of the model and observer operators around the current mean in order to directly reuse the LKF. The posterior probability density (resulting in the transformation of Gaussian probability density with mean \bar{X} and covariance matrix \mathbf{C}_x by the nonlinear operator \mathcal{A}) with mean \bar{Y} and covariance matrix \mathbf{C}_y is thus obtained as:

$$\begin{cases} \bar{Y} = \mathcal{A}(\bar{X}) \\ \mathbf{C}_y = \mathbf{A}^T \mathbf{C}_x \mathbf{A} \quad \mathbf{A} = \nabla \mathcal{A} \end{cases} \quad (5.4)$$

Nevertheless, the accuracy of this approach is not enough when the nonlinearities cannot be properly linearized (Mariani & Ghisi, 2007; Diaz, 2023).

Ensemble Kalman Filter (EnKF): Monte-Carlo sampling and propagation

To overcome these limitations, the statistical linearization technique can be used (Gelb, 1974; Lefebvre et al., 2004). In this technique, a nonlinear function of a random variable is linearized through a linear regression between sampling points drawn from the prior distribution of the random variable. This type of method has led to the Ensemble Kalman Filter introduced in (Evensen, 1994). In this case, the posterior probability density is obtained with:

$$\begin{cases} \bar{Y} = \frac{1}{N} \sum_{i=1}^N \mathcal{A}(X_i) \\ \mathbf{C}_y = \frac{1}{N} \sum_{i=1}^N (\mathcal{A}(X_i) - \bar{Y})(\mathcal{A}(X_i) - \bar{Y})^T \end{cases} \quad (5.5)$$

with $X_i \sim \mathcal{N}(\bar{X}, \mathbf{C}_x)$

The limitation of this approach is the important number of sampled points required to obtain the posterior probability density. Indeed, the model evaluation can be computationally expensive, thus making this approach not suited for sequential data assimilation.

Unscented Kalman Filter (UKF): use of a limited number of σ -points

The Unscented Kalman Filter (UKF) (Julier & Uhlmann, 1997) was introduced to decrease the number of sampled points, called σ -points. It requires the sampling of $2L + 1$ σ -points (where L is the size of the state vector).

The idea, based on the so-called Unscented transform, is to deterministically sample well-chosen σ -points and propagate them through the nonlinear operator. The σ -points are computed with the expressions presented in Table 5.1. The numerically efficient Cholesky factorization method is generally used to calculate the matrix square root. The posterior probability density is then obtained with:

$$\begin{cases} \bar{Y} = \frac{1}{N} \sum_{i=1}^{2L+1} \omega_i^m \mathcal{A}(X_i) & X_i = \bar{X} \pm [\sqrt{\mathbf{C}_x} \mathbf{S}]_i \\ \mathbf{C}_y = \frac{1}{N} \sum_{i=1}^{2L+1} \omega_i^c (\mathcal{A}(X_i) - \bar{Y})(\mathcal{A}(X_i) - \bar{Y})^T \end{cases} \quad (5.6)$$

where ω_i^m and ω_i^c are weights associated with the σ -points, and \mathbf{S} a matrix used for the definition of the σ -points.

Scaled Spherical Simplex Filter (S3F): UKF with less σ -points

Recently introduced in (Amir et al., 2022; Papakonstantinou et al., 2022), the Scaled Spherical Simplex Filter (S3F) requires $L+2$ σ -points, thus reducing by 50% the computation

time. The posterior probability density is obtained in the same way than (5.6), the only difference being the number of σ -points.

The developments of this chapter are based on the S3F because of this computational efficiency. Algorithm 4 presents the S3F algorithm (as well as the UKF because they share the same basics, only the definition of σ -points differs).

Algorithm 4 UKF/S3F

- 1: **Input:**
 - 2: Observations z_k , model and observation operators \mathcal{M} , \mathcal{H} , noise covariance matrices \mathbf{Q} for model bias and \mathbf{R} for measurement noise, S3F parameters: (α, β, κ) , number of state parameters L
 - 3: Initial state estimation \hat{x}_0 and its associated covariance matrix \mathbf{C}_0
 - 4:
 - 5: **Initialization:**
 - 6: Computation of the constant weights associated with σ -points (according to Table 5.1):
 - 7: Computation of the constant matrix \mathbf{S} used for the definition of σ -points (according to Table 5.1):
 - 8:
 - 9: **for** $k = 1 : \infty$ **do**
 - 10: $\mathcal{X}_{k,i} = \hat{x}_{k-1} + (\sqrt{\mathbf{C}_{k-1}}\mathbf{S})_i \quad \forall i \in \llbracket 1, N \rrbracket$ ▷ Calculating N σ -points
 - 11:
 - 12: **Prediction:**
 - 13: $\hat{\mathcal{X}}_{k,i}^- = \mathcal{M}(\mathcal{X}_{k,i}) \quad \forall i \in \llbracket 1, N \rrbracket$ ▷ Prediction for each σ -point
 - 14: $\hat{x}_k^- = \sum_{i=1}^N \omega_i^m \hat{\mathcal{X}}_{k,i}^-$ ▷ Weighted mean to predict the state
 - 15: $\mathbf{C}_k^- = \sum_{i=1}^N \omega_i^c (\hat{\mathcal{X}}_{k,i}^- - \hat{x}_k^-)(\hat{\mathcal{X}}_{k,i}^- - \hat{x}_k^-)^T + \mathbf{Q}$ ▷ Associated covariance matrix
 - 16:
 - 17: $\mathcal{Z}_{k,i} = \mathcal{H}(\mathcal{X}_{k,i}) \quad \forall i \in \llbracket 1, N \rrbracket$ ▷ Observation for each σ -point
 - 18: $\hat{z}_k^- = \sum_{i=1}^N \omega_i^m \mathcal{Z}_{k,i}$ ▷ Weighted mean for the observations
 - 19: $\mathbf{C}_{zz} = \sum_{i=1}^N \omega_i^c (\mathcal{Z}_{k,i} - \hat{z}_k^-)(\mathcal{Z}_{k,i} - \hat{z}_k^-)^T + \mathbf{R}$ ▷ Covariance
 - 20: $\mathbf{C}_{xz}^- = \sum_{i=1}^N \omega_i^c (\hat{\mathcal{X}}_{k,i}^- - \hat{x}_k^-)(\mathcal{Z}_{k,i} - \hat{z}_k^-)^T$ ▷ Cross-covariance
 - 21:
 - 22: **Correction:**
 - 23: $\mathbf{K}_k = \mathbf{C}_{xz}^- \mathbf{C}_{zz}^{-1}$ ▷ Kalman gain
 - 24: $\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k(z_k - \hat{z}_k^-)$ ▷ Updated state
 - 25: $\mathbf{C}_k = \mathbf{C}_k^- - \mathbf{K}_k \mathbf{C}_{zz} \mathbf{K}_k^T$ ▷ Associated updated covariance
 - 26: **end for**
-

Remark

For the sake of simplicity, Algorithm 4 is presented in the case where the noise is assumed to be additive. One can refer to (Julier & Uhlmann, 1997) to see a more general framework.

Joint Kalman Filter

The first approach consists of using only one Kalman Filter in which the model parameters are seen as components of the same vector as the state. To do so, the parameter vector is concatenated with the state vector to produce a joint state $x_k^* = [x_k^T \ p_k^T]^T$, leading to the Joint Kalman Filter framework:

$$\begin{cases} x_k^* &= \begin{bmatrix} \mathcal{M}_{k-1}(x_{k-1}, p_{k-1}, w_{x,k-1}) \\ p_k = p_{k-1} + w_{p,k-1} \end{bmatrix} \\ z_k &= \mathcal{H}_k(x_k, v_k) \end{cases} \quad (5.8)$$

Dual Kalman Filter

In the Dual Kalman Filter approach, the state vector is only composed of the parameters to identify. The observer operator is converted into a state evaluation operator \mathcal{H}_{dual} , based on a second Kalman Filter (5.10), accounting for the evolution of the dynamical system. The denomination "dual" comes from the use of two Kalman Filters.

$$\begin{cases} p_k = p_{k-1} + w_{p,k-1} \\ z_k = \mathcal{H}_{dual}(x_k, p_k, w_{x,k}, v_k) \end{cases} \quad (5.9)$$

$$\begin{cases} x_k = \mathcal{M}_{k-1}(x_{k-1}, p_{k-1}, w_{x,k-1}) \\ z_k = \mathcal{H}(x_k, p_k, w_{x,k}, v_k) \end{cases} \quad (5.10)$$

Both of these approaches have been widely used for nonlinear identification applications by various nonlinear extensions of Kalman filter. The Dual Kalman Filter approach might be more costly, but can provide better estimates (Mariani & Corigliano, 2005). The following of this Chapter builds on the Dual Kalman Filter approach in a modified version that integrates nonlinear KF extension and the mCRE framework.

2 Extension of MDKF to address evolution laws

As already mentioned in this chapter introduction, KF is suited for sequential data assimilation but is not robust to high measurements noise. On the contrary, the mCRE is very robust to noisy measurements but is not suited for sequential data assimilation. The mCRE and the Dual Kalman Filter approaches were coupled in (Marchand et al., 2016) in the context of unsteady thermal applications. By doing so, this coupling overcomes the limitations of both mCRE and Kalman Filters, resulting in a fast and robust sequential model updating method called the Modified Dual Kalman Filter (MDKF). In (Diaz et al., 2023a) the MDKF was further modified, and verified for low-frequency dynamics cases by applying the developed framework on two earthquake engineering examples. The present section

extends the works of (Marchand et al., 2016; Diaz et al., 2023a) to the case of nonlinear history-dependent material behavior.

Algorithm 5 details the MDKF approach in the context of nonlinear history-dependent material behavior. The key idea of the MDKF approach is to use the admissible fields of the mCRE obtained from the trade-off between the model and observations at the end of Step 1 (see Section 1.1.1) as the dual observations in the Dual Kalman Filter. Contrary to the use of a classical projection matrix as an observer to collect the measurements to compare to the predictions, the use of the mCRE admissible fields offers high robustness to noise. Indeed, in line 17, for each σ -point the Step 1 of the mCRE minimization is performed in order to compute the admissible fields resulting in a compromise between model and observations. The parameters are updated at each step searching for the values giving a zero gradient of mCRE (see line 24 in Algorithm 5).

Algorithm 5 MDKF

- 1: **Input:**
 - 2: Model and observation operators \mathcal{M} , \mathcal{H} , noise covariance matrices \mathbf{Q} for model bias, KF parameters: $(\alpha_{KF}, \beta, \kappa)$,
 - 3: Initial state estimation \hat{x}_0 and its associated covariance matrix \mathbf{C}_0
 - 4:
 - 5: **Initialization:**
 - 6: Computation of the constant weights associated with σ -points (according to Table 5.1):
 - 7: Computation of the constant matrix \mathbf{S} used for the definition of σ -points (according to Table 5.1):
 - 8:
 - 9: **for** $k = 1 : \infty$ **do**
 - 10: $\mathcal{X}_{k,i} = \hat{p}_{k-1} + (\sqrt{\mathbf{C}_{k-1}}\mathbf{S})_i \quad \forall i \in \llbracket 1, N \rrbracket$ ▷ Calculating N σ -points
 - 11:
 - 12: $\mathcal{X}_{k,i}^- = \mathcal{X}_{k,i} \quad \forall i \in \llbracket 1, N \rrbracket$ ▷ Prediction for each σ -point
 - 13: $\hat{p}_k^- = \sum_{i=1}^N \omega_i^m \mathcal{X}_{k,i}^-$ ▷ Weighted mean to predict the parameters
 - 14: $\mathbf{C}_k^- = \sum_{i=1}^N \omega_i^c (\mathcal{X}_{k,i}^- - \hat{p}_k^-)(\mathcal{X}_{k,i}^- - \hat{p}_k^-)^T + \mathbf{Q}$ ▷ Associated covariance matrix
 - 15:
 - 16: For each σ -point, perform mCRE Step 1 to compute admissible fields and set:
 - 17: $\mathcal{Z}_{k,i} = \nabla_p \mathcal{E}_{mCRE}(\mathcal{X}_{k,i}^-)$
 - 18: $\hat{z}_k = \sum_{i=1}^N \omega_i^m \mathcal{Z}_{k,i}$ ▷ Weighted mean for the observations
 - 19: $\mathbf{C}_{zz} = \sum_{i=1}^N \omega_i^c (\mathcal{Z}_{k,i} - \hat{z}_k)(\mathcal{Z}_{k,i} - \hat{z}_k)^T$ ▷ Covariance
 - 20: $\mathbf{C}_{xz}^- = \sum_{i=1}^N \omega_i^c (\mathcal{X}_{k,i}^- - \hat{p}_k^-)(\mathcal{Z}_{k,i} - \hat{z}_k)^T$ ▷ Cross-covariance
 - 21:
 - 22: **Correction:**
 - 23: $\mathbf{K}_k = \mathbf{C}_{xz}^- \mathbf{C}_{zz}^{-1}$ ▷ Kalman gain
 - 24: $\hat{p}_k = \hat{p}_k^- + \mathbf{K}_k(0 - \hat{z}_k^-)$ ▷ By replacing $z_k = 0$, the parameters are searched to have a zero mCRE gradient.
 - 25: $\mathbf{C}_k = \mathbf{C}_k^- - \mathbf{K}_k \mathbf{C}_{zz} \mathbf{K}_k^T$ ▷ Associated updated covariance
 - 26: **end for**
-

git Online tool: Two 1D-examples can be found in the repository of this thesis: one for the updating of a Young modulus (without evolution law on the material behavior) and one for the updating of an isotropic hardening modulus (with evolution law). See `Chap_5/MDKF_elastic` and `Chap_5/MDKF_evolution`.

3 Results

This section aims to illustrate the MDKF method on an elastoplastic case. As mentioned in this chapter introduction, the goal of this chapter is not to conduct an extensive analysis of the performance of the MDKF method, but only to check that the MDKF framework can be easily adapted to the case of history-dependent constitutive model. A two-dimensional beam loaded in tension is considered. The considered beam has the same geometry and sensor position as in Figure 3.9. The constitutive model considered is the following:

- The free energy ψ is split into the elastic part ψ_e and plastic part ψ_p with

$$\psi_e(\epsilon_e) = \frac{1}{2}(\lambda(\text{tr } \epsilon_e)^2 + 2\mu \epsilon_e : \epsilon_e) \quad \text{where } \lambda = \frac{E \nu}{(1 + \nu)(1 - 2\nu)}, \mu = \frac{E}{2(1 + \nu)} \quad (5.11)$$

$$\psi_p(\epsilon_p) = \frac{1}{2} h p^2 \quad (5.12)$$

- The dissipation pseudo-potential is:

$$\varphi^*(f) = \mathbf{I}(f) \quad \text{where } f = \sigma_{eq} - (R + R_0) \text{ is the Von Mises Yield criterion} \quad (5.13)$$

with $\mathbf{I}(f)$ the indicator function, E the Young modulus, ν the Poisson ratio, R_0 the initial Yield limit and h the isotropic hardening modulus.

This model is used to generate the synthetic data representing optic fiber observations. A white Gaussian noise of 0,01% is then added to these observations. In the following the form of this model is assumed to be known. For the sake of simplicity, all the parameters are assumed to be known except the isotropic hardening modulus which is updated. The values of the parameters are the same as in Chapter 3 in Table 3.4. 500 loading steps are considered and Figure 5.4 shows the evolution of the estimated normalized isotropic hardening modulus as a function of time. This figure shows that the hardening modulus is properly identified as the normalized middle σ -point converges close to the true value. The stationarity of the middle σ -point at the beginning (until $t = 250$) is explained by the fact that the material is still in the elasticity domain, yet the hardening modulus is not sensitive.

4 Toward a coupling between MDKF and model bias correction with neural networks

The previous section presented the parameter identification of a given model form. Therefore this method is potentially subject to model bias when the assumed model is too far

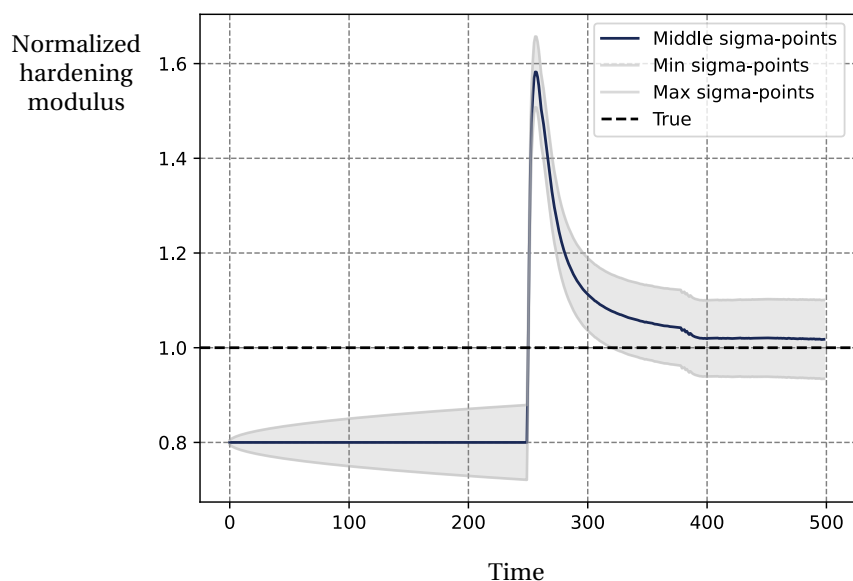


Figure 5.4 • Evolution of the normalized isotropic hardening modulus as a function of the time, for the 3 σ -points.

from reality. Yet, using the MDKF procedure to update the neural network parameters is too costly because the number of σ -points directly depends on the number of parameters to identify. S3F requires $L + 2$ σ -points and UKF requires $2L + 1$ σ -points (where L is the number of parameters to identify).

An idea to correct the model bias while allowing for a continuous sequential data assimilation is summarized in Figure 5.5. This idea, which has not been implemented yet, relies on a continuous dialog between the model bias correction procedure (developed in Chapters 2 and 3), with the MDKF-based sequential data assimilation. The key idea is to use the modeling error term available in the mCRE at each state estimation in order to evaluate the quality of the predictions. When the value of the modeling error is too high, the model bias correction procedure can be performed, based on all the historical data available at this time. This model bias correction can be performed with three major steps.

First, the neural networks can be initialized with the best model available (in the sense of CRE modeling error). This model can come from a catalog of physical models or from a previously trained neural network. This step thus involves a model selection based on the CRE value as well as a supervised neural network training when the selected model is taken from the catalog of physical models.

The second step consists of a mCRE minimization with the historical data available at this time. This procedure might be very costly, but does not need to be performed quickly as the sequential data assimilation can still be performed in parallel. If one wants to reduce the cost of this minimization, it is possible to sample only relevant timesteps for this training. The relevant timesteps can be, once again, chosen with a criterion based on the CRE term evaluated on the MDKF estimated states: to correct model bias it is relevant to use the

timestep with a high modeling error.

Finally, the trained network cannot be directly used in the MDKF because of the high number of parameters. This motivates the need for postprocessing the trained network to sparsify it. (Fuhg et al., 2024) recently proposed a sparsification method suited for input-convex neural networks in the context of model discovery. In this approach, sparse and interpretable expressions are recovered from the neural networks, with a smoothed version of L^0 -regularization. The obtained expression is parametrized, and these parameters can be sequentially updated in the MDKF procedure.

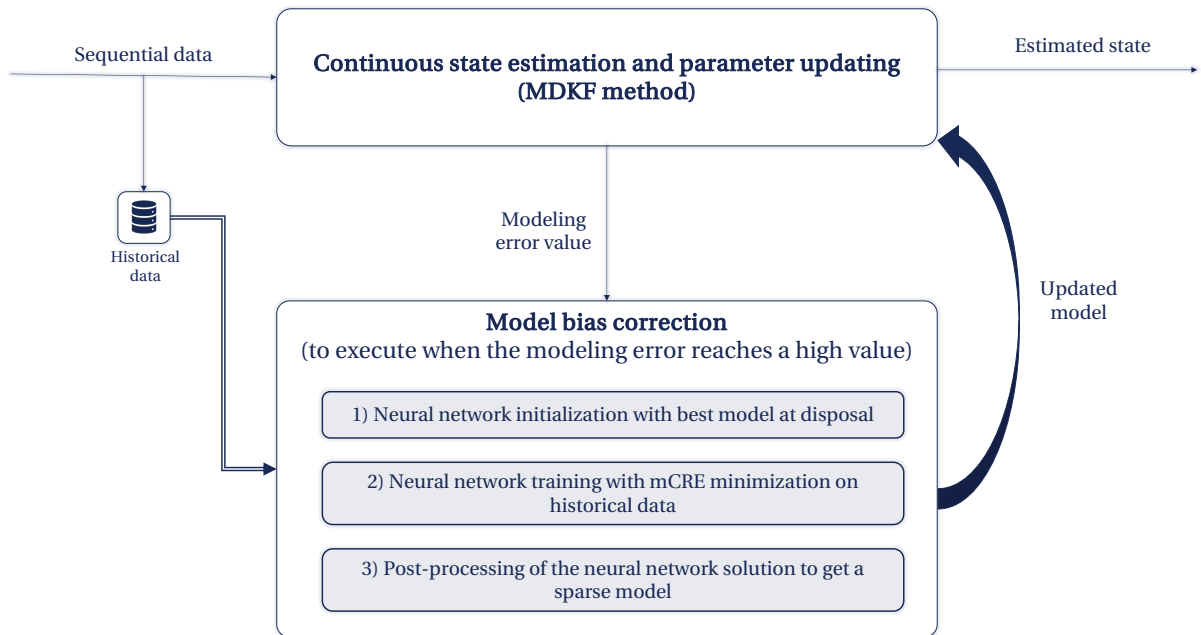


Figure 5.5 • Strategy to correct model bias along the structure lifecycle.

Chapter conclusion

In this chapter, we presented a Modified Dual Kalman Filtering (MDKF) approach aimed at sequential state estimation and parameter identification. This chapter checked that the previous works on MDKF can be extended to integrate the case where the material behavior involves nonlinear evolution laws. This method integrates the recent development in Kalman Filtering approaches, especially the Scaled Spherical Simplex Filter (S3F) which offers accurate results with a small number of σ -points, even in the presence of important nonlinearities. By integrating on-the-fly observations with the robust mCRE framework, the MDKF offers an efficient and reliable estimation of evolving parameters that is robust to noise.

The method has been illustrated with an elastoplastic behavior in which an isotropic hardening modulus was sequentially updated. This preliminary work shows promising results on a relatively simple 2D case, which paves the way for the sequential updating of more complex constitutive models after further investigations. Nevertheless, at this stage, no reduced order modeling strategy has been implemented which does not make this approach suited for real-time data assimilation. The method should be further investigated with evolutive model parameters, coupled with the NN-mCRE approach for model bias correction, and tested with real experimental data.

Conclusion and perspectives

Main contributions of the thesis

The main objective of this thesis was to develop advanced numerical methods for model bias correction, allowing accurate prediction of material behavior throughout the structural life cycle. To achieve this, the key idea was to free the model form by using neural networks to represent constitutive models, which allows the model form not to be assumed a priori.

This choice of using neural networks to represent constitutive models without a pre-defined form raised several critical questions. How can the neural network be effectively trained to model constitutive behavior without direct stress-strain data? How can the model ensure compliance with established physical laws and reliable knowledge? How can the reliability of the learned models be quantitatively assessed to ensure their applicability in practical scenarios?

The mCRE framework has proven to be very helpful in answering these questions. In fact, this manuscript has shown that this framework is suitable for training neural networks in an unsupervised manner. With the mCRE, it is natural to formulate the constitutive models with thermodynamic potentials using the Generalized Standard Material framework. Adherence to this framework was ensured by using input-convex neural networks to represent convex thermodynamic potentials. The other part of reliable knowledge, the static and kinematic admissibility, is enforced in the minimization process by computing admissible fields before each model update. Finally, the CRE term provides a rich physical meaning that is used as a modeling error to assess the reliability of the model.

The developed method was implemented in different cases. Chapter 2 focused on the model bias correction of history-independent behavior such as nonlinear elastic and hyperelastic behavior. In Chapter 3, history-dependent dissipative behaviors were considered with an emphasis on elastoplasticity and viscoplasticity.

Neural network training is known to be very sensitive to the choice of user-defined hyperparameters, which motivated the development of automatic hyperparameter tuning. Without careful selection of the network architecture, training may converge to a local minimum. Chapter 4 addressed this issue by introducing a one-layer input-convex neural network that is less subject to local minima.

Finally, the development of the mCRE framework to deal with nonlinear evolution laws has been used in Chapter 5 to extend the MDKF approach, an mCRE-based Kalman Filter method suited for sequential data assimilation.

While significant advancements have been made regarding the model bias correction, several areas require further exploration to enhance the applicability and efficiency of the methods.

Prospect: dealing with unknown variables

In this manuscript, the internal variables were assumed to be known. In practice, the knowledge of the number of internal variables might be difficult to determine and may be an important source of modeling error. In the approach developed in this manuscript, no model bias correction for the number of internal variables was implemented, which is a strong limitation prior to application with real data. Learning internal variables is therefore a natural perspective to continue this work. There are a couple of works in the literature that aim at learning internal variables (Rosenkranz et al., 2024; Bonatti & Mohr, 2021), but to the author's knowledge, they all are performed in a supervised way (with strain-stress data). A research direction could be to use the architecture (based on recurrent neural networks) proposed in the supervised contributions and train them in an unsupervised manner.

Prospect: efficiency of the minimization process through coupling with reduced order modeling

The examples treated in this thesis were limited to two-dimensional geometries, with relatively simple loading cases. The computational cost was already significant (a few hours for a 2D geometry with history-dependent behavior). The efficiency of the method should therefore be improved before treating real cases with complex geometry and loading history, such as in the target of the DREAM-ON project.

Coupling the proposed method with reduced order modeling (ROM) techniques is therefore a relevant research direction. Among the wide range of ROM methods, the Proper Generalized Decomposition (PGD) (Chinesta et al., 2011, 2014) consists in iteratively constructing, on-the-fly, a low-rank representation of the solution from a modal decomposition with separated variables. This decomposition, computed by solving eigenvalue problems, allows to obtain an approximate parametric solution that explicitly depends on all model parameters (boundary conditions, geometry, material properties, etc.) considered as extra-coordinates, and then describes the solution of all possible scenarios. Recently, approaches have been proposed to apply PGD to nonlinear models, such as the LATIN-PGD method (Cremonesi et al., 2013; Relun et al., 2013) using the iterative LATIN method (Ladevèze, 1999). The LATIN-PGD method has already been coupled with the mCRE in (Marchand et al., 2016). It would be interesting to update this work in the case where the constitutive model is described by neural networks. Another approach to integrating ROM techniques is to rely on iterative proper orthogonal decomposition (POD) using approaches similar to (Peherstorfer & Willcox, 2015a; Haasdonk et al., 2023).

Prospect: evaluation of the method on experimental data

In this manuscript, only synthetic data were considered for the mCRE minimization or the MDKF-based sequential data assimilation. A white Gaussian noise was added to the synthetic data, which may be different from the real-life noise. The evaluation of the developed method should then be performed with real-life data. Currently, work is in progress to train neural networks within the mCRE framework on the experimental data presented in the Appendix B (where neural networks are trained with another unsupervised method on digital image correlation data).

Prospect: assembling the blocks of the DREAM-ON project to control the structural health

Finally, the different parts of the DREAM-ON project (model bias correction, sequential data assimilation, model predictive control) should be assembled to build in real-time a hybrid twin, used to control the structural health to limit its damage. The pioneering ideas concerning the coupling between sequential data assimilation and model bias correction (including a sparsification of the model) were established in Chapter 5, Section 4, but the implementation remains a perspective. The use of the updated model for model predictive control is also an open question.

Acknowledgements

This thesis has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101002857).

Appendices

A

Comparison between NN-EUCLID and NN-mCRE

This appendix is a summary of the Master thesis of Edgar Zembra that I supervised during the summer 2023. The reference of the open-access master thesis of Edgar Zembra is: Edgar Zembra, Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Use of physics-augmented neural networks for unsupervised learning of material constitutive relations - Comparison of the NN-Euclid and NN-mCRE methods. ENS Paris-Saclay; Centrale Supélec. 2023. <https://hal.science/hal-04255767>

The internship of Edgar Zembra was motivated by the need for a comparison between the methods of unsupervised neural network training suited to deal with measurable data. To the best of the author's knowledge, NN-EUCLID and NN-mCRE are currently the only two approaches in the literature to treat the unsupervised training of neural networks for constitutive modeling. In this appendix, the NN-EUCLID method is briefly described in Section 1 and is compared with the NN-mCRE approach.

1 EUCLID framework in brief

The idea of the NN-EUCLID method (Thakolkaran et al., 2022) is to train a neural network to minimize the equilibrium gap (Claire et al., 2004). A strain field, computed from a measured displacement field, is used as the input of a neural network. The neural network outputs a stress field that is then used to compute the loss function based on the equilibrium gap. The neural network parameters are updated with a gradient descent until convergence.

The following presents this process more formally. Consider displacement field data $\mathbf{u}(\mathbf{x})$ defined on spatial positions \mathbf{x} . This displacement field may come from DIC/DVC measurement procedures and is supposed to be known on the sample surface. The displacement field is expressed using a finite element mesh, composed of N_n nodes, with finite

element shape functions $\Phi^i(\mathbf{x})$ such that:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{N_n} \mathbf{u}_e^i \Phi^i(\mathbf{x}), \quad (\text{A.1})$$

with \mathbf{u}_e^i being the nodal displacement amplitudes. The deformation gradient field is then approximated as follows:

$$\mathbf{F}(\mathbf{x}) = \mathbf{I} + \sum_{i=1}^{N_n} \mathbf{u}_e^i \nabla \Phi^i(\mathbf{x}), \quad (\text{A.2})$$

where \mathbf{I} is the identity matrix and ∇ the gradient operator defined in Lagrangian specifications.

Let a specimen be loaded with N_R reaction forces R_i with $i \in [1, N_R]$, obtained with external force sensors. The EUCLID framework consists of minimizing a loss function based on the force balance residuals given by a first term \mathcal{L}_{int} at all free nodes and a second term \mathcal{L}_{BC} at the controlled boundary conditions. The N_n nodes of the finite element mesh can hence be divided into an inner part N_{int} and the controlled boundary condition part N_{BC} , with $N_n = N_{int} + N_{BC}$.

In case of negligible body forces, the nodal residual can be written by integrating the stresses onto the reference domain Ω :

$$f_i = \int_{\Omega} P_{ij} \nabla_j \Phi^i dV = \int_{\Omega} \mathcal{M}[\mathbf{F}]_{ij} \nabla_j \Phi^i dV. \quad (\text{A.3})$$

Where \mathcal{M} is a constitutive model mapping deformation \mathbf{F} onto first Piola-Kirchhoff stress $\mathbf{P}(\mathbf{F}) = \mathcal{M}[\mathbf{F}]$.

The EUCLID framework proposes a first loss term minimizing the L2 norm on the nodal force residuals (ensuring a balanced state).

$$\mathcal{L}_{int}(t) = \sum_{i=1}^{N_{int}} [f_i(t)]^2, \quad (\text{A.4})$$

The second term expects to balance the reaction forces

$$\mathcal{L}_{BC}(t) = \sum_{j=1}^{N_R} \left[R_j(t) - \sum_{i=1}^{N_{BC}} f_i(t) \right]^2. \quad (\text{A.5})$$

The complete loss consists of a weighted sum of the two terms described above, with λ a weighting factor (Flaschel et al., 2021). The loss also sums all N_t temporal contributions such that:

$$\mathcal{L}_l = \frac{1}{N_t} \sum_t [\hat{\mathcal{L}}_{int}(t) + \lambda \mathcal{L}_{BC}(t)]. \quad (\text{A.6})$$

The EUCLID method has been extended with the learning of neural-network constitutive models and is referred to as NN-EUCLID (Thakolkaran et al., 2022).

2 Comparison of NN-mCRE and NN-EUCLID

This section aims to summarize the comparison between the NN-mCRE and the NN-EUCLID approaches. The main difference comes from the choice of the loss function: an equilibrium-based loss for NN-EUCLID, and a loss function based on the constitutive relation error for NN-mCRE. The choice of the loss function has many implications on the optimization process, the possibilities offered by the methods, and the performance.

It is common to present the mCRE framework through the spectrum of information reliability. The philosophy of the mCRE is therefore to strongly enforce reliable information while releasing unreliable information. An efficient way to understand the conceptual differences between the two approaches is to compare what is considered reliable and what is not. First, both approaches release the model form (unreliable) using neural networks while ensuring the thermodynamic consistency (reliable) through input-convex neural networks. The fundamental difference lies in the consideration of the measured displacement field. In NN-EUCLID the measured field is used directly to compute the stress field, thus considering the measurements as reliable. On the contrary, NN-mCRE considers the measurements as unreliable because of the noise: the displacement fields used to compute the gradient of the loss function are thus obtained from the computation of admissible fields (Step 1 of the mCRE minimization). By doing so, the admissibility is considered reliable in NN-mCRE, whereas it is only penalized in the loss function in NN-EUCLID.

Remark

The CRE for inverse problem method (in its "not modified" version) used to consider the measurements as reliable by integrating the observations in the definition of the kinematic admissibility (Ladevèze & Reynier, 1989). This approach was not suitable for dealing with high noise levels, which motivated the development of the mCRE.

That being said, the two methods exhibit different features regarding the computational cost, the robustness to noise, and the possibility of dealing with incomplete observations. An interested reader should refer to the master thesis of Edgar Zembra for the quantitative comparison.

Computational cost

From the experiments conducted during Edgar Zembra's internship, NN-EUCLID demonstrated a higher computation efficiency. This observation is explained by the fact that NN-EUCLID is not required to compute admissible fields before each gradient step. Indeed the computational cost of the mCRE comes from the need to compute admissible fields before each gradient step: the computation of admissible fields requires the assembly and inversion of the tangent matrix.

Noise robustness

NN-mCRE showed as expected a better robustness to noise than the NN-EUCLID frame-

work. This is explained by the use by NN-EUCLID of the measured displacement fields directly to compute the stress field used for the loss function: the noise thus propagates from the displacement to the loss function, resulting in inaccurate model update in presence of high noise. Nevertheless, the author thinks that this limitation can be overcome through the development of a method accounting for noise propagation. Furthermore, with the value of noise generally obtained with digital image correlation, this lack of noise robustness is handlable.

Incomplete observations

The last point of comparison concerns the possibility of treating incomplete observations, such as strain observations coming from optical fibers. This is one strong advantage of NN-mCRE over NN-EUCLID because NN-EUCLID cannot treat partial measurements. Indeed, to compute the equilibrium gap, it is necessary to have a stress field everywhere in the structure. As the stress field is computed with the constitutive model directly from the strain field obtained from displacement measurements, it is not possible to compute a stress field everywhere with partial measurements.

B

Learning a hyperelastic behavior with the NN-EUCLID method applied on experimental data

This appendix is a summary of a contribution to the unsupervised training of Physics-Augmented neural networks (PANN) with the NN-EUCLID method applied to real data. This work was made in collaboration with Clément Jailin, Rémi Legroux and Emmanuel Baranger and was subject to a submission in *Experimental Mechanics*: Experimental learning of a hyperelastic behavior with a Physics Augmented Neural Network.

Nowadays, the current approaches in the literature train the PANN from synthetic data. The developments are hence restricted to numerical proof of concepts. The noise included in the studies differs from real experimental noise (*e.g.* sometimes applied on the deformation or displacement fields) and other sensor uncertainty (*e.g.* force sensor axis uncertainty). The quantity of data used in synthetic approaches can be tuned to train the model successfully. Finally, procedures are rarely blinded; by knowing the targeted model (knowing the shape of the free energy surface, for example), the neural network architecture can hence be adequately designed using the required deformation invariants. The closest study to the current development in the literature is (Li & Chen, 2022), where the authors validate their AI model from DIC measurements in a compression test on rubber cubes. However, besides being a different constitutive model (without physical knowledge), the experiment is limited to small deformations (under 5%). Multiple questions are still open for the experimental training of highly constrained PANN models, architecture selection, and model validation. The identification of PANN from actual experimental data hence remains a challenge and has not been studied in the literature beyond 1D applications (Jordan et al., 2020; Diamantopoulou et al., 2021; Pierre et al., 2023; Flaschel et al., 2021).

Figure B.1 summarizes the contribution detailed in this appendix. Here, a PANN is

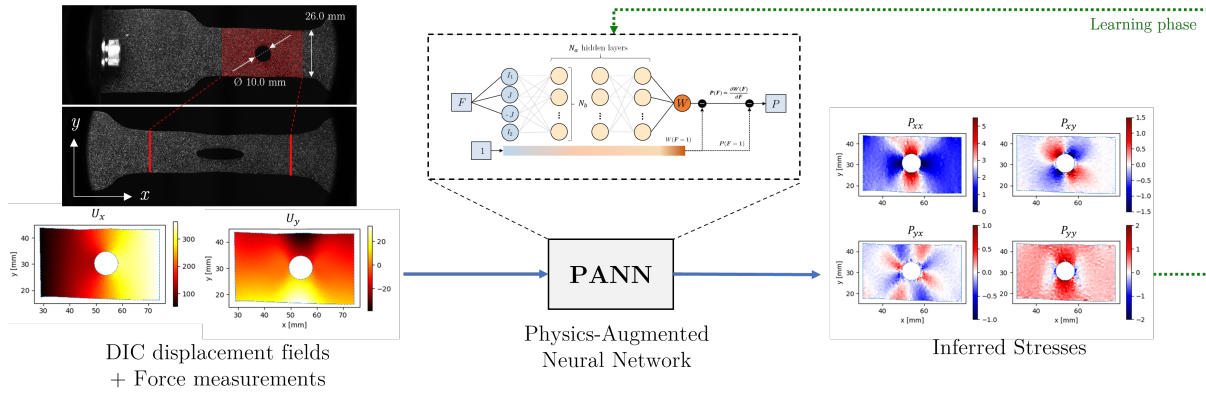


Figure B.1 • Summary of the implemented approach

trained and tested within an NN-EUCLID (see Appendix A for a brief description of NN-EUCLID) framework based on experiments performed on hyperelastic behavior with digital image correlation (DIC) measurements. The results of the training are compared with a traditional Neo-Hookean model (with identified parameters within the EUCLID framework). In this model variant, the strain energy density function is written with the deformation invariants defined in plane stress, with $I_1 = \text{tr}(\mathbf{C})$ and $J = \det(\mathbf{F})$, with $\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}$:

$$W_{NH} = \frac{\mu}{2} \cdot (I_1 - 2) - \mu \cdot \log(J) + \frac{\lambda}{2} \cdot \log(J)^2, \quad (\text{B.1})$$

with μ and λ the Lamé coefficients.

The network used in this application is an input-convex neural network, similar to the one used in this dissertation. A study on the architecture has been implemented (not detailed here) to find the number of layers and neurons which enable to best fit the observations.

Two uni-axial tensile tests have been performed on a rubber-like hyperelastic material with unknown a priori properties. A single sample was loaded twice, first with a full geometry and a second time after a hole was created and the sample repositioned. In the second test, the shape of the sample was cut with a hole to enhance a heterogeneous loading and increase the mechanical content of the measurement. The loaded surface was imaged using a PCO-edge camera with a telecentric lens (diameter 125 mm) and LED panels. At this scale, each pixel has a resolution of $78 \mu\text{m}$. (electronic noise estimated standard deviation: $\sigma_{f1} = 374$ gray levels in experiment-1, and $\sigma_{f2} = 340$ gray levels in experiment-2 for 16-bit GL images). The uni-axial testing machine was a thermo-regulated electro-mechanical INSTRON system equipped with a 5 kN loading cell (estimated standard deviation $\sigma_R = 0.020$ N). The sample was continuously loaded during the test with a displacement control. The tensile displacement measured by the machine is written v . The material was initially cycled to reach a stabilized behavior. Figure B.2 shows the DIC results in terms of displacement and strain for one timestep.

The imaged surface was covered with thin white paint speckles to create an image gradient used for the DIC measurement. As the sample color was initially black, no black

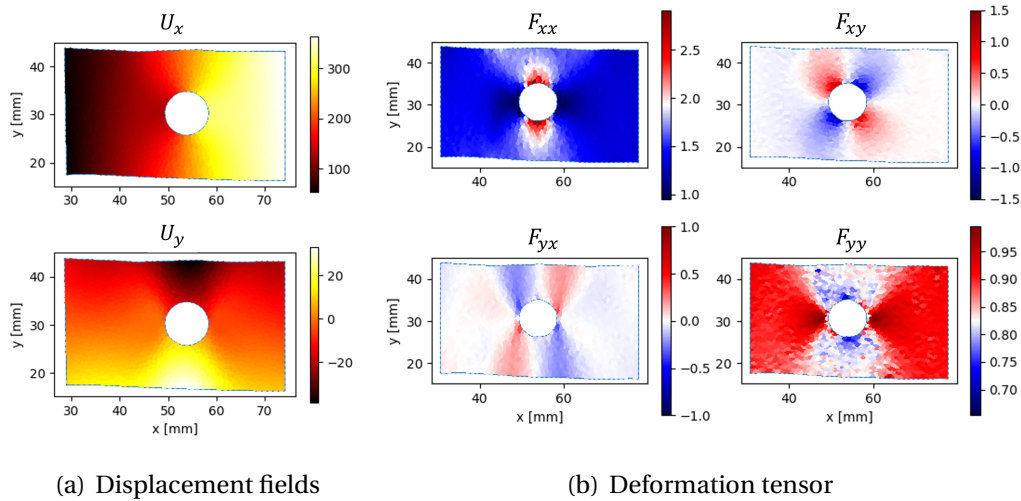


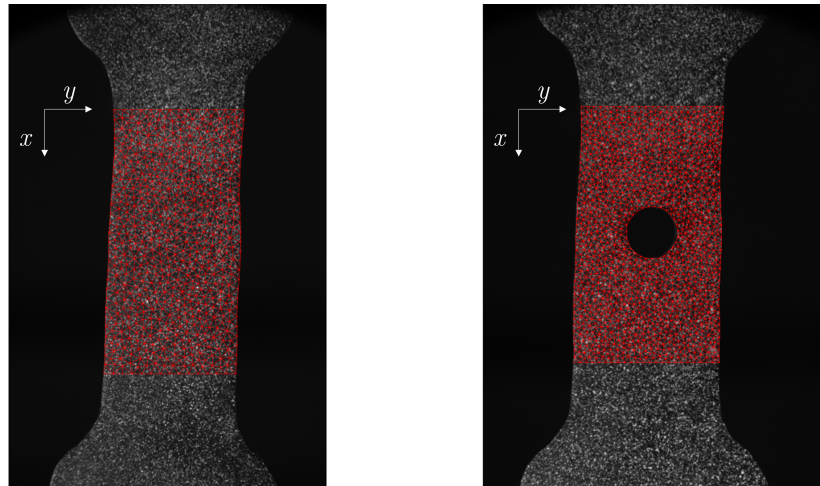
Figure B.2 • DIC results for one timestep: (a) displacement field, expressed in pixels and (b) deformation tensor.

paint was used (allowing for a reduction in the projected paint quantity and thus reducing the risk of paint cracking at large deformation). The shapes and appearances of the samples are shown in figure B.3(a).

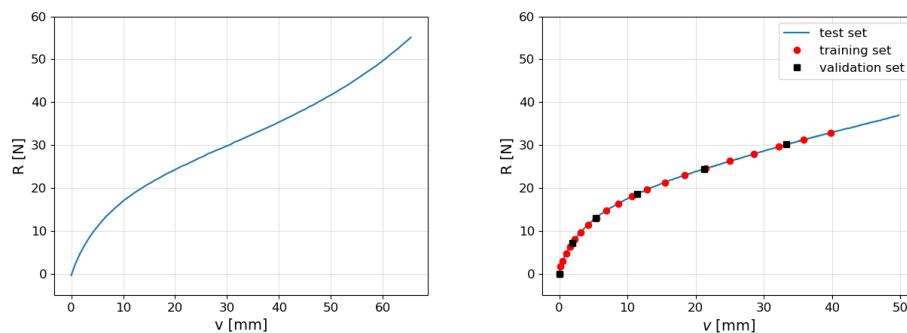
The PANN was trained during 20,000 epochs of the NN-EUCLID procedure. The loss starts at values around 10^4 and converges to 121. For comparison, the loss estimated on the 30-first unloaded DIC computation gives 4.2 (std 0.22). The EUCLID procedure was also applied to identify the two material parameters of the NH model. At convergence, the obtained parameters were $E_{\text{NH}} = 1.74$ MPa and $\nu_{\text{NH}} = 0.471$. The final converged value of the loss was 299. This equilibrium loss is much higher than the previous PANN results.

The equilibrium metrics at every loading step of the test sets for both models are presented in Figure B.4 and B.5, respectively, for the internal metrics and boundary condition metric. First, the metrics evolve smoothly, without significant discontinuity between the steps close to the trained ones. This suggests that the model does not completely overfit the trained data and is able to generalize the behavior. At the end of experiment-2, some steps show discontinuous high values for both models when the loading is important. As the NH model is also impacted, the reason comes from the input deformation fields. Inaccurate DIC measurements pollute those steps and may correspond to outlier deformation element values. The first outliers in the inner part appear at around 35 mm and at the boundary condition at around 50 mm.

For experiment-1, the internal loss is low for both models (under 50). Most of the total loss is represented by the boundary condition part. For experiment-2, the internal part increases with the loading. After approximately 31.5 mm (thus after the training loads), the PANN internal error becomes higher than the NH one. This may highlight an issue in extrapolating after the trained steps. However, the boundary condition loss part shows lower values for the PANN model compared to the NH model. Finally, the last steps (around 50 mm) show important boundary condition metrics value. It may also be due to inaccurate



(a) Geometries and DIC meshes for experiment-1 (left) and experiment-2 (right)



(b) Force displacement curves for experiment-1 (left) and experiment-2 (right)

Figure B.3 • Samples, meshes, and force/displacement curves measured by the loading machine for experiment-1 (a) and experiment-2 (b). The red circles in (b) indicate the 20 training data and the black squares represent the validation data. All other blue steps are used for the test.

DIC measurement on nodes at the boundary condition, as it happens for both the PANN and NH models.

Conclusion

This study presents the application of a thermodynamics-augmented neural network, PANN, learning 2D real hyperelastic behavior. Through the use of two uni-axial experiments, instrumented by digital image correlation and force sensors, the PANN model was trained within an EUCLID framework and validated. The experiments, which achieved axial deformations of over 200% and showcased marked non-linear behavior, provided robust training and validation datasets. 20 loading steps from one experiment were employed for training, while an extensive set of 640 loading steps, from both experiments, was used for validation. Comparative analysis of our model with a covariance-weighted error highlighted the PANN approach's capability in understanding material behavior, in particular when extrapolating beyond the training load amplitudes, indicating the model's robustness and predictive accuracy. This research not only highlights the potential of neural networks

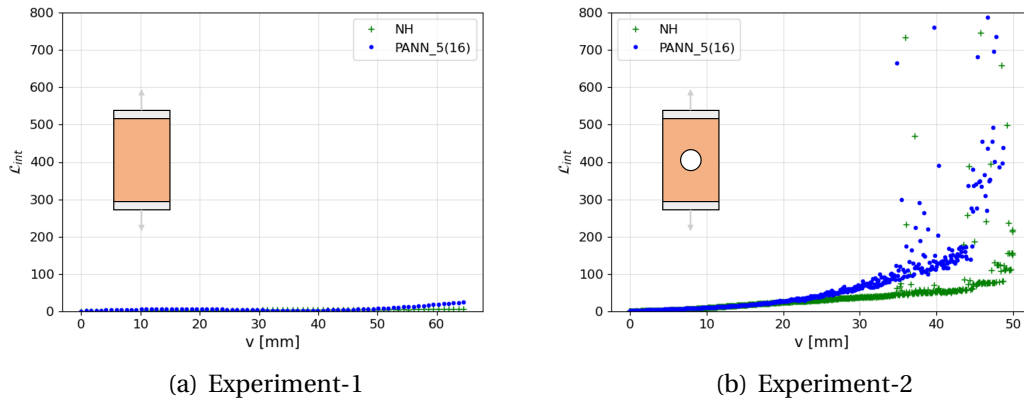


Figure B.4 • Internal metric, \mathcal{L}_{int} , evaluated on the test sets for the different models: PANN_5(16) and NH.

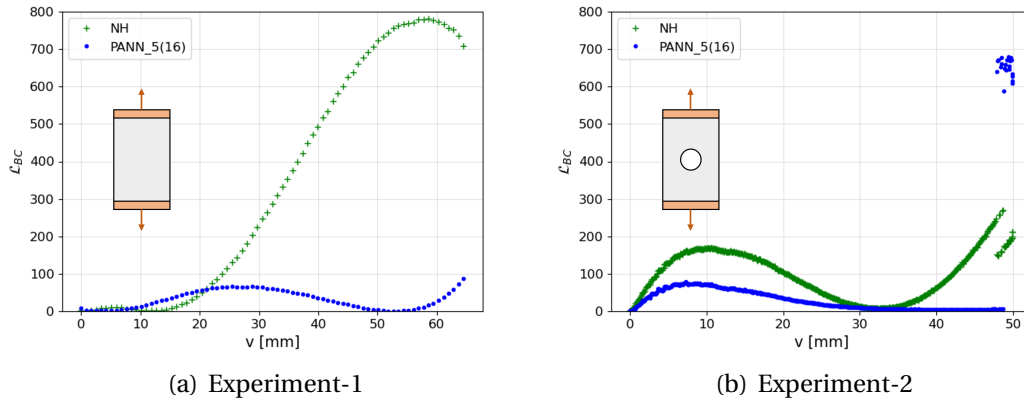


Figure B.5 • Boundary condition metric, \mathcal{L}_{BC} , evaluated on the test sets for the different models: PANN_5(16) and NH.

in modeling material behavior but also opens the way for experimental neural network constitutive models in materials science. The full article also includes discussions on the model selection, the mechanical contents of the training and validation datasets, as well as more a more detailed comparison between the NH and PANN models.

C

Repository with the Python tutorials

To support the concepts and methodologies discussed throughout this dissertation, I have created a comprehensive Python tutorial available on GitHub. This repository includes step-by-step instructions, code samples, and relevant datasets designed to enhance understanding and facilitate practical application. To access the repository, please visit https://github.com/Antoine-benady/thesis_tutorial

The content of the repository is the following:

- The folder "Chapter_1" contains a 1D example for the mCRE minimization, in the case of the identification of a Young modulus.
- The folder "Chapter_3" illustrates the mCRE minimization involving an elastoplastic behavior, in 1D, for the identification of an isotropic hardening modulus.
- The folder "Chapter_5" shows two implementations of the MDKF approach, in 1D, one for the updating of a Young modulus (history-independent behavior) and one for the updating of an isotropic hardening modulus (history-dependent behavior).

D

Publications and communications

The research work presented in this manuscript has been the subject of publications and communications in conferences that are listed below:

Publications in peer-reviewed journals

Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. NN-mCRE: a modified Constitutive Relation Error framework for unsupervised learning of nonlinear state laws with physics-augmented Neural Networks. *International Journal for Numerical Methods in Engineering*, 2024, 125(8):e7439. <https://dx.doi.org/10.1002/nme.7439>

Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Unsupervised learning of history-dependent constitutive material laws with thermodynamically-consistent neural networks in the modified Constitutive Relation Error framework. *Computer Methods in Applied Mechanics and Engineering*, 2024, 425, pp.116967. <https://dx.doi.org/10.1016/j.cma.2024.116967>

Clément Jailin, Antoine Benady, Rémi Legroux and Emmanuel Baranger. Experimental learning of a hyperelastic behavior with a Physics Augmented Neural Network. *Experimental Mechanics (under review)*.

Antoine Benady, Sahar Farahbakhsh, Martin Poncelet, Emmanuel Baranger, and Ludovic Chamoin. A modified dual Kalman Filter approach for damage model updating. (*In preparation*).

International communications

Ludovic Chamoin, Antoine Benady, Sahar Farahbakhsh, Emmanuel Baranger, Martin Poncelet. Adaptive modeling and learning of material laws for effective data assimilation. 16th World Congress on Computational Mechanics, Jul 2024, Vancouver, Canada.

Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Physics-Augmented Neural Networks for Constitutive Modeling: Toward an Application for Structural Health Moni-

toring. The 9th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS Congress 2024, Jun 2024, Lisbon, Portugal.

Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Physics-augmented neural networks for constitutive modeling: training with the modified Constitutive Relation Error. MORTech 2023 – 6th International Workshop on Model Reduction Techniques, Nov 2023, Gif-sur-Yvette, France.

Antoine Benady, Ludovic Chamoin, Emmanuel Baranger. A modified Constitutive Relation Error framework to learn nonlinear constitutive laws using physics-augmented Neural Networks. IACM Mechanistic Machine Learning and Digital Engineering for Computational Science Engineering and Technology., Sep 2023, El Paso, United States.

Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. A modified Constitutive Relation Error (mCRE) framework to learn nonlinear constitutive models from strain measurements with thermodynamics-consistent Neural Networks. XI Conference on Adaptive Modeling and Simulation, Jun 2023, Gothenburg, Sweden.

Antoine Benady Ludovic Chamoin, Emmanuel Baranger. Physics-informed neural networks derived from a mCRE functional for constitutive modeling. IUTAM Symposium on Data-driven Mechanics, Oct 2022, Paris, France.

Ludovic Chamoin, Antoine Benady, Sahar Farahbakhsh, Emmanuel Baranger, Martin Poncelet. Data-based Model Updating, Selection, and Enrichment using the Modified Constitutive Relation Error Concept. 15th World Congress on Computational Mechanics, Jul 2022, Yokohama, Japan

National communications

Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Apprentissage non-supervisé de lois de comportement nonlinéaires avec réseau de neurones thermodynamiquement consistant par minimisation de l'erreur en relation de comportement modifiée. 16ème Colloque National en Calcul des Structures, Mai 2024, Presqu'île de Giens, France.

Antoine Benady, Ludovic Chamoin, Emanuel Baranger. Physics-informed neural networks derived from a mCRE functional for constitutive modeling. Artificial Intelligence and Augmented Engineering, Dec 2022, Palaiseau, France.

Antoine Benady, Ludovic Chamoin, Emmanuel Baranger. Réseaux de neurones informés par la physique pour l'apprentissage de lois de comportement.. 25ème Congrès Français de Mécanique 2022, Aug 2022, Nantes, France.

Antoine Benady, Ludovic Chamoin, Emmanuel Baranger. Réseaux de neurones infor-

més par la physique pour l'apprentissage de lois de comportement. IA pour les sciences de l'ingénierie, Jun 2022, Online, France.

Book chapter

Ludovic Chamoin, Emmanuel Baranger, Antoine Benady, Pierre-Étienne Charbonnel, Matthieu Diaz, Sahar Farahbakhsh, Laurent Fribourg, Daniel Martin Xavier, Martin Poncelet. A novel DDDAS architecture combining advanced sensing and simulation technologies for effective real-time structural health monitoring. 2024. <https://hal.science/hal-04425882> (*To be published*).

Supervised master internship

Edgar Zembra, Antoine Benady, Emmanuel Baranger, Ludovic Chamoin. Use of physics-augmented neural networks for unsupervised learning of material constitutive relations - Comparison of the NN-Euclid and NN-mCRE methods. ENS Paris-Saclay; Centrale Supélec. 2023. <https://hal.science/hal-04255767>

E

Extended abstract in french

Cette annexe comprend un résumé étendu du manuscrit, rédigé en français.

Le contrôle de santé des structures demeure une préoccupation essentielle en ingénierie, étant donné les impératifs de sécurité et de durabilité. Une pratique récente consiste à mettre en place un dialogue entre une structure physique et son jumeau numérique, afin de prédire l'état de santé et de limiter l'endommagement. C'est l'objectif du projet ERC DREAM-ON, dans lequel s'inscrit cette thèse, réalisée au Laboratoire de Mécanique Paris-Saclay (LMPS). Celle-ci vise à construire le jumeau numérique d'une structure physique de manière à obtenir un modèle prédictif de la santé de la structure afin de limiter son endommagement via des lois de contrôle. Au sein du projet DREAM-ON, cette thèse s'intéresse à la construction automatique d'une loi de comportement décrite par un réseau de neurones à partir de données mesurables. Le manuscrit s'articule autour de cinq chapitres encadrés par une introduction et une conclusion.

L'introduction précise le cadre de l'étude et le contexte lié au projet ERC DREAM-ON. L'idée développée dans ce travail est d'utiliser les réseaux de neurones dont les applications récentes sont très nombreuses, couplés à la technique de l'erreur en relation de comportement modifiée qui est une technique largement étudiée dans le cadre de problèmes inverses.

Le premier chapitre correspond à une synthèse bibliographique qui met en place les outils de base utilisés, portant d'une part sur l'erreur en relation de comportement modifiée, et d'autre part sur les réseaux de neurones en se focalisant sur leur application dans le champ de la physique. La revue de littérature sur l'erreur en relation de comportement modifiée a permis d'identifier une limite des précédents travaux : la forme de la loi de comportement est systématiquement postulée a priori, laissant alors la méthode d'identification sujet à un biais de modèle. L'étude bibliographique sur les réseaux de neurones est alors justifiée par la possibilité offerte par les réseaux de neurones de représenter n'importe quelle fonction continue (théorème d'approximation universelle), permettant ainsi de s'affranchir

du biais de modèle lorsque le réseau de neurones décrit la loi de comportement. Après avoir présenté les réseaux de neurones et les techniques de couplage avec la physique, cette bibliographie s'intéresse à l'utilisation récente des réseaux de neurones pour décrire des lois de comportement matériau, notamment sur l'utilisation de réseaux de neurones validant par construction les principes de la thermodynamique. Une limite actuelle de la littérature concerne l'entraînement « non-supervisé » de lois de comportement dans la mesure où la plupart des travaux utilisent des données de contraintes, non mesurables en pratique. La conclusion de ce chapitre invite donc à entraîner les réseaux de neurones en minimisant la fonctionnelle d'erreur en relation de comportement modifiée, permettant ainsi de traiter le problème d'apprentissage non-supervisé.

Dans le deuxième chapitre, une méthode de couplage entre les réseaux de neurones et l'erreur en relation de comportement modifiée est proposée pour traiter les problèmes d'élasticité et d'hyperélasticité. Cette procédure s'appuie sur l'utilisation de réseaux de neurones convexes pour décrire un potentiel thermodynamique. Les paramètres du réseau sont cherchés de manière à minimiser l'erreur en relation de comportement modifiée via une procédure d'optimisation spécifiquement développé pour le cas où le comportement est décrit par réseau de neurones. Une discussion importante est portée sur les différentes manières de réduire la sensibilité des hyperparamètres choisis. Différents cas tests viennent illustrer l'efficacité de l'algorithme proposé, notamment sur la robustesse au bruit de mesure, la qualité des solutions apprises, et la faible sensibilité aux choix des hyperparamètres.

Le troisième chapitre complète le chapitre précédent en étendant l'idée proposée aux comportements matériaux dépendant de l'histoire, ce qui nécessite des adaptations pour l'algorithme de minimisation liées à la nécessité d'intégrer les lois d'évolution. La méthode est d'abord illustrée sur un cas unidimensionnel de plasticité avec loi d'érouissage isotrope (associé à un tutoriel python en accès libre), puis sur l'apprentissage de lois élastoplastique et viscoplastique. Les conclusions du précédent chapitre sont confirmées sur les différents cas tests.

Le chapitre 4 propose d'étudier plus en détail l'architecture du réseau de neurones. Dans ce chapitre, l'idée est d'étudier le réseau lui-même et de le rendre le plus efficace possible par rapport à son utilisation, notamment pour réduire au maximum la présence de minima locaux. Les choix des fonctions d'activation, du nombre de couches, de l'initialisation sont donc étudiés, afin de proposer une paramétrisation qui s'avère particulièrement efficace pour ce problème. Cette paramétrisation est intéressante dans la mesure où elle ne repose que sur une seule couche cachée, ce qui va à l'encontre de nombreux développements récents en apprentissage profond. Ceci s'explique par le fait que les fonctions à apprendre sont convexes, et donc relativement simple comparés aux fonctions apprises en apprentissage profond. Cette paramétrisation à une couche offre par ailleurs l'avantage de comprendre certaines choses en termes d'extrapolation, bien que cette observation reste à explorer sur des cas plus difficiles.

Dans le chapitre 5, un autre challenge du projet est abordé ; il s'agit de l'assimilation de données séquentielles. Ce chapitre est relativement exploratoire, car différentes idées

sont présentées, mais elles ne sont qu'une première pierre à l'objectif initial proposé dans l'introduction. L'idée est fondée sur le couplage entre l'erreur en relation de comportement modifiée et les filtres de Kalman déjà développé au laboratoire. Ce chapitre consiste à étendre les précédents travaux de manière à intégrer dans ce cadre les comportements matériaux non-linéaires et dépendant de l'histoire.

Enfin, la conclusion dresse un bilan des travaux réalisés dans ce manuscrit. Au niveau des perspectives, plusieurs pistes sont identifiées et classées. Les perspectives concernent principalement l'apprentissage des variables internes et la rapidité de la méthode avec le couplage avec des outils de réduction de modèle. Des annexes présentent également certains travaux menés en périphérie de la thèse, avec notamment l'utilisation d'une autre méthode de la littérature (NN-EUCLID) sur des données expérimentales et la comparaison de cette dernière avec la méthode développée en thèse.

Bibliography

- Abisset, E. (2012).** *Un mésomodèle d'endommagement des composites stratifiés pour le virtual testing : identification et validation* (Theses 2012DENS0035). École normale supérieure de Cachan - ENS Cachan. (Cit. on p. 2).
- Abueidda, D. W., Koric, S., Sobh, N. A., & Sehitoglu, H. (2021).** Deep learning for plasticity and thermo-viscoplasticity. *International Journal of Plasticity*, 136, 102852 (cit. on p. 34).
- Aitken, A. C. (1935).** Note on Selection from a Multivariate Normal Population. *Proceedings of the Edinburgh Mathematical Society*, 4(2), 106–110 (cit. on p. 110).
- Aldakheel, F., Elsayed, E. S., Zohdi, T. I., et al. (2023).** Efficient multiscale modeling of heterogeneous materials using deep neural networks. *Computational Mechanics*, 72, 155–171 (cit. on p. 31).
- Allix, O., Feissel, P., & Nguyen, H. M. (2005).** Identification strategy in the presence of corrupted measurements. *Engineering Computations*, 22(5/6), 487–504 (cit. on pp. 5, 15, 16, 59, 80).
- Allix, O., Feissel, P., & Thévenet, P. (2003).** A delay damage mesomodel of laminates under dynamic loading: Basic aspects and identification issues. *Computers and Structures*, 81(12), 1177–1191 (cit. on p. 15).
- Amir, M., Papakonstantinou, K. G., & Warn, G. P. (2022).** Scaled Spherical Simplex Filter and State-Space Damage-Plasticity Finite-Element Model for Computationally Efficient System Identification. *Journal of Engineering Mechanics*, 148(2), 04021138 (cit. on pp. 110, 115).
- Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016).** Deep speech 2: End-to-end speech recognition in english and mandarin, 173–182 (cit. on p. 21).
- Amos, B., Xu, L., & Kolter, J. Z. (2017).** Input convex neural networks In *Proceedings of the 34th international conference on machine learning*. PMLR. 146–155. (Cit. on pp. 29, 32, 70, 96, 98).
- Andrieux, S., Abda, A., & Bui, H. (1999).** Reciprocity principle and crack identification. *Inverse Problems*, 15(1), 59–65 (cit. on p. 5).

- Aquino, W., & Bonnet, M. (2019).** Analysis of the error in constitutive equation approach for time-harmonic elasticity imaging. *SIAM Journal on Applied Mathematics*, 79, 822–849 (cit. on p. 16).
- As'ad, F., Avery, P., & Farhat, C. (2022).** A mechanics-informed artificial neural network approach in data-driven constitutive modeling. *International Journal for Numerical Methods in Engineering*, 123(12), 2738–2759 (cit. on p. 33).
- Banerjee, B., Walsh, T., Aquino, W., & Bonnet, M. (2013).** Large scale parameter estimation problems in frequency-domain elastodynamics using an error in constitutive equation functional. *Computer Methods in Applied Mechanics and Engineering*, 253, 60–72 (cit. on p. 15).
- Barbarella, E., Allix, O., Daghia, F., Lamon, J., & Jollivet, T. (2016).** A new inverse approach for the localization and characterization of defects based on compressive experiments. *Computational Mechanics*, 57(6), 1061–1074 (cit. on pp. 15, 16).
- Barrault, M., Maday, Y., Nguyen, N. C., & Patera, A. T. (2004).** An empirical interpolation method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9), 667–672 (cit. on p. 2).
- Barthe, D., Ladevèze, P., Deraemaeker, A., & Loch, S. L. (2004).** Validation and updating of industrial models based on the constitutive relation error. *AIAA Journal*, 42, 1427–1434 (cit. on p. 15).
- Ben Azzouna, M., Feissel, P., & Villon, P. (2015).** Robust identification of elastic properties using the modified constitutive relation error. *Computer Methods in Applied Mechanics and Engineering*, 295, 196–218 (cit. on pp. 15, 16).
- Bertino, L., Evensen, G., & Wackernagel, H. (2003).** Sequential data assimilation techniques in oceanography. *International Statistical Review / Revue Internationale de Statistique*, 71(2), 223–241 (cit. on p. 110).
- Bischof, R., & Kraus, M. (2022).** Multi-objective loss balancing for physics-informed deep learning (cit. on pp. 28, 46).
- Bishara, D., Xie, Y., Liu, W., & Li, S. (2023).** A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials. *Archives of Computational Methods in Engineering*, 30, 191–222 (cit. on p. 30).
- Bishop, C. M. (2007).** *Pattern recognition and machine learning (information science and statistics)*. Springer-Verlag. (Cit. on p. 21).
- Blasch, E. P., Darema, F., & Bernstein, D. (2022).** Introduction to the dynamic data driven applications systems (dddas) paradigm. In *Handbook of dynamic data driven applications systems: Volume 1* (pp. 1–32). Springer International Publishing. (Cit. on p. 3).

- Boehler, J. P. (1987).** Introduction to the invariant formulation of anisotropic constitutive equations. In *Applications of tensor functions in solid mechanics* (pp. 13–30). Springer Vienna. (Cit. on p. 31).
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016).** End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (cit. on p. 21).
- Bonatti, C., & Mohr, D. (2021).** One for all: Universal material model based on minimal state-space neural networks. *Science Advances*, 7(26), eabf3658 (cit. on pp. 34, 126).
- Bonatti, C., & Mohr, D. (2022).** On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids. *Journal of the Mechanics and Physics of Solids*, 158, 104697 (cit. on p. 34).
- Bonnet, M., & Constantinescu, A. (2005).** Inverse problems in elasticity. *Inverse Problems*, 21 (cit. on pp. 5, 16, 34).
- Bonnet, M., & Aquino, W. (2015).** Three-dimensional transient elastodynamic inversion using an error in constitutive relation functional. *Inverse Problems*, 31(3), 035010 (cit. on pp. 15, 16).
- Bonnet, M., Salasiya, P., & Guzina, B. B. (2024).** Error-in-constitutive-relation (ecr) framework for the characterization of linear viscoelastic solids. *Working paper or preprint* (cit. on p. 64).
- Bono, F. M., Radicioni, L., Cinquemani, S., Benedetti, L., Cazzulani, G., Somaschini, C., & Belloli, M. (2023).** A deep learning approach to detect failures in bridges based on the coherence of signals. *Future Internet*, 15(4) (cit. on p. 31).
- Bouclier, R., Louf, F., & Chamoin, L. (2013).** Real-time validation of mechanical models coupling pgd and constitutive relation error. *Computational Mechanics*, 52(4), 861–883 (cit. on p. 15).
- Boyd, S., & Vandenberghe, L. (2004).** *Convex optimization*. Cambridge university press. (Cit. on p. 43).
- Breiman, L. (2001).** Random forests. *Machine learning*, 45(1), 5–32 (cit. on p. 5).
- Bucy, R., & Senne, K. (1971).** Digital synthesis of non-linear filters. *Automatica*, 7(3), 287–298 (cit. on p. 110).
- Bui, H., & Constantinescu, A. (2000).** Spatial localization of the error of constitutive law for the identification of defects in elastic bodies. *Archives of Mechanics*, 52(4-5), 511–522 (cit. on pp. 6, 15, 16).
- Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. (2021).** Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12), 1727–1738 (cit. on p. 27).

- Chaboche, J. (1988).** Continuum damage mechanics: Part i - general concepts. *Journal of Applied Mechanics*, 55, 59–64 (cit. on p. 2).
- Chaboche, J. (1993).** Development of continuum damage mechanics for elastic solids sustaining anisotropic and unilateral damage. *International Journal of Damage Mechanics*, 2(4), 311–329 (cit. on p. 2).
- Chamoin, L., Florentin, E., Pavot, S., & Visseq, V. (2012).** Robust goal-oriented error estimation based on the constitutive relation error for stochastic problems. *Computers and Structures*, 106-107, 189–195 (cit. on p. 19).
- Chamoin, L., & Ladevèze, P. (2008).** A non-intrusive method for the calculation of strict and efficient bounds of calculated outputs of interest in linear viscoelasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 197(9-12), 994–1014 (cit. on p. 19).
- Chamoin, L., Pled, F., Allier, P.-E., & Ladevèze, P. (2017).** A posteriori error estimation and adaptive strategy for pgd model reduction applied to parametrized linear parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 327, 118–146 (cit. on p. 19).
- Chamoin, L. (2021).** DREAM-ON: Merging advanced sensing techniques and simulation tools for future structural health monitoring technologies. In *The Project Repository Journal* (pp. 124–127). (Cit. on pp. 1, 3, 58).
- Chamoin, L., Allier, P.-E., & Marchand, B. (2016).** Synergies between the constitutive relation error concept and pgd model reduction for simplified v&v procedures. *Advanced Modeling and Simulation in Engineering Sciences*, 3 (cit. on p. 90).
- Chamoin, L., Farahbakhsh, S., & Poncelet, M. (2022).** An educational review on distributed optic fiber sensing based on rayleigh backscattering for damage tracking and structural health monitoring. *Measurement Science and Technology*, 33(12), 124008 (cit. on p. 2).
- Charbonnel, P.-E., Ladevèze, P., Louf, F., & Noach, C. L. (2013).** A robust cre-based approach for model updating using in situ measurements. *Computers and Structures*, 129, 63–73 (cit. on p. 15).
- Chen, C., Seff, A., Kornhauser, A., & Xiao, J. (2015).** Deepdriving: Learning affordance for direct perception in autonomous driving In *Proceedings of the ieee international conference on computer vision*. IEEE. 2722–2730. (Cit. on p. 21).
- Chesné, S., & Deraemaeker, A. (2013).** Damage localization using transmissibility functions: A critical review. *Mechanical Systems and Signal Processing*, 38, 569–584 (cit. on p. 2).
- Chinesta, F., Keunings, R., & Leygue, A. (2014).** *The proper generalized decomposition for advanced numerical simulations*. Springer International Publishing. (Cit. on pp. 2, 126).

- Chinesta, F., Ladevèze, P., & Cueto, E. (2011).** A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18, 395–404 (cit. on pp. 2, 126).
- Chinesta, F., Cueto, E. G., Abisset-Chavanne, E., Duval, J. L., & Khaldi, F. E. (2019).** Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. *Archives of Computational Methods in Engineering*, 27, 105–134 (cit. on p. 3).
- Chouaki, A., Ladevèze, P., & Proslie, L. (1996).** An updating of structural dynamic model with damping. *Inverse Problems in Engineering: Theory and Practice*, 335–342 (cit. on p. 15).
- Chouaki, A., Ladevèze, P., & Proslie, L. (1998).** Updating structural dynamic models with emphasis on the damping properties. *AIAA Journal*, 36(6), 1094–1099 (cit. on p. 15).
- Claire, D., Hild, F., & Roux, S. (2004).** A finite element formulation to identify damage fields: The equilibrium gap method. *International Journal for Numerical Methods in Engineering*, 61(2), 189–208 (cit. on pp. 5, 35, 131).
- Cohen, T. S., & Welling, M. (2016).** Group equivariant convolutional networks In *Proceedings of the 33rd international conference on international conference on machine learning - volume 48*. New York, NY, USA, JMLR.org. 2990–2999. (Cit. on p. 29).
- Coorevits, P., Ladevèze, P., Pelle, J., & Rougeot, P. (1992).** Some new applications of a method for the control and optimization of finite element computations. *New advances in computational structural mechanics*, 205–217 (cit. on p. 9).
- Cortes, C., & Vapnik, V. (1995).** Support-vector networks. *Machine Learning*, 20(3), 273–297 (cit. on p. 5).
- Cottreau, R., Díez, P., & Huerta, A. (2009).** Strict error bounds for linear solid mechanics problems using a subdomain based flux-free method. *Computational Mechanics*, 44(4), 533–547 (cit. on p. 10).
- Cremonesi, M., Néron, D., Guidault, P., & Ladevèze, P. (2013).** A pgd-based homogenization technique for the resolution of nonlinear multiscale problems. *Computer Methods in Applied Mechanics and Engineering*, 267, 275–292 (cit. on p. 126).
- Cybenko, G. (1989).** Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4), 303–314 (cit. on p. 24).
- Darema, F. (2015).** Dddas, a key driver for large-scale-big-data and large-scale-big-computing In *Iccs 2015* (S. Koziel, L. Leifsson, M. Lees, V. Krzhizhanovskaya, J. Dongarra, & P. Sloot, Eds.). 2463. (Cit. on p. 3).

- Decouvreur, V., Bouillard, P., Deraemaeker, A., & Ladevèze, P. (2004).** Updating 2d acoustic models with the constitutive relation error method. *Journal of Sound and Vibration*, 278(4-5), 773–787 (cit. on p. 15).
- Decouvreur, V., Deraemaeker, A., Ladevèze, P., & Bouillard, P. (2007).** Building a suited reduced modal basis for updating 3d acoustic models with the constitutive law error method. *Computer Methods in Applied Mechanics and Engineering*, 196(35-36), 3400–3408 (cit. on p. 15).
- Decouvreur, V., Ladevèze, P., & Bouillard, P. (2008).** Updating 3d acoustic models with the constitutive relation error method: A two-stage approach for absorbing material characterization. *Journal of Sound and Vibration*, 310, 985–997 (cit. on p. 15).
- Del Moral, P. (1996).** Non linear filtering: Interacting particle solution. *Markov Processes and Related Fields*, 2, 555–580 (cit. on p. 110).
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009).** Imagenet: A large-scale hierarchical image database In *2009 ieee conference on computer vision and pattern recognition*. 248–255. (Cit. on p. 21).
- Deraemaeker, A., Ladevèze, P., & Leconte, P. (2002).** Reduced bases for model updating in structural dynamics based on constitutive relation error. *Computer Methods in Applied Mechanics and Engineering*, 191(21), 2427–2444 (cit. on p. 15).
- Deraemaeker, A., Ladevèze, P., & Romeuf, T. (2004).** Model validation in the presence of uncertain experimental data. *Engineering Computations*, 21, 808–833 (cit. on pp. 6, 15, 16, 46).
- Destuynder, P., & Métivet, B. (1999).** Explicit error bounds in a conforming finite element method. *Mathematics of Computation*, 68(228), 1379–1396 (cit. on p. 8).
- Diamantopoulou, M., Karathanasopoulos, N., & Mohr, D. (2021).** Stress-strain response of polymers made through two-photon lithography: Micro-scale experiments and neural network modeling. *Additive Manufacturing*, 47, 102266 (cit. on p. 135).
- Diaz, M., Charbonnel, P.-E., & Chamoin, L. (2023a).** A new kalman filter approach for structural parameter tracking: Application to the monitoring of damaging structures tested on shaking-tables. *Mechanical Systems and Signal Processing*, 182, 109529 (cit. on pp. 111, 118, 119).
- Diaz, M., Charbonnel, P., & Chamoin, L. (2024).** Fully automated model updating framework for damage detection based on the modified constitutive relation error. *Computational Mechanics*, 73, 619–638 (cit. on p. 15).
- Diaz, M. (2023).** *Numerical framework for data-driven model-based monitoring of dynamical systems - Application to earthquake engineering* (Theses 2023UPAST096). Université Paris-Saclay. (Cit. on pp. 16, 110, 115, 117).

- Diaz, M., Charbonnel, P.-E., & Chamoin, L. (2022).** Robust energy-based model updating framework for random processes in dynamics: Application to shaking-table experiments. *Computers and Structures*, 264 (cit. on pp. 15, 46).
- Diaz, M., Charbonnel, P.-E., & Chamoin, L. (2023b).** A new physics-guided data assimilation framework for online structural monitoring: Application to shaking-table tests. Athens, Greece. 4217–4233. (Cit. on p. 111).
- Diaz, M., Aquino, W., & Bonnet, M. (2015).** A modified error in constitutive equation approach for frequency-domain viscoelasticity imaging using interior data. *Computer Methods in Applied Mechanics and Engineering*, 296, 129–149 (cit. on pp. 6, 15, 16).
- Doebling, S. W., Farrar, C. R., Prime, M. B., & Shevitz, D. W. (1996).** *Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review* (tech. rep. LA-13070-MS). Los Alamos National Laboratory. (Cit. on p. 1).
- Doebling, S., Farrar, C., & Prime, M. (1998).** A summary review of vibration-based damage identification methods. *Shock & Vibration Digest*, 30(2), 91–105 (cit. on p. 2).
- Dornheim, J., Morand, L., Nallani, H. J., et al. (2024).** Neural networks for constitutive modeling: From universal function approximators to advanced models and the integration of physics. *Archives of Computational Methods in Engineering*, 31, 1097–1127 (cit. on p. 5).
- Duchi, J., Hazan, E., & Singer, Y. (2011).** Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12, 2121–2159 (cit. on p. 26).
- Ern, A., Nicaise, S., & Vohralik, M. (2007).** An accurate $h(\text{div})$ flux reconstruction for discontinuous galerkin approximations of elliptic problems. *Comptes Rendus Académie des Sciences, Paris, Série I*, 345(12), 709–712 (cit. on p. 10).
- Ern, A., & Vohralik, M. (2010).** A posteriori error estimation based on potential and flux reconstruction for the heat equation. *SIAM Journal on Numerical Analysis*, 48(1), 198–223 (cit. on p. 10).
- Evensen, G. (1994).** Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5), 10143–10162 (cit. on pp. 110, 115).
- Evensen, G. (2003).** The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4), 343–367 (cit. on p. 110).
- Fang, Z. (2022).** A high-efficient hybrid physics-informed neural networks based on convolutional neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10), 5514–5526 (cit. on p. 29).

- Farahani, A., Pourshojae, B., Rasheed, K., & Arabnia, H. R. (2020).** A concise review of transfer learning In *2020 international conference on computational science and computational intelligence (csci)*. 344–351. (Cit. on p. 29).
- Farrar, C. R., & Worden, K. (2007).** An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 303–315 (cit. on p. 1).
- Faverjon, B., Ladevèze, P., & Louf, F. (2009).** Validation of stochastic linear structural dynamics models. *Computers & Structures*, 87(13-14), 829–837 (cit. on p. 15).
- Faverjon, B., & Sinou, J.-J. (2008).** Robust damage assessment of multiple cracks based on the frequency response function and the constitutive relation error updating method. *Journal of Sound and Vibration*, 312, 821–837 (cit. on p. 15).
- Feissel, P., & Allix, O. (2007).** Modified constitutive relation error identification strategy for transient dynamics with corrupted data: The elastic case. *Computer Methods in Applied Mechanics and Engineering*, 196(13), 1968–1983 (cit. on pp. 5, 15, 16, 59, 80).
- Ferrier, R., Cocchi, A., & Hochard, C. (2021).** Modified constitutive relation error for field identification: Theoretical and experimental assessments on fiber orientation identification in a composite material. *International Journal for Numerical Methods in Engineering*, 122, 7553–7580 (cit. on pp. 15, 16).
- Fish, J. (2011).** Multiscale modeling and simulation of composite materials and structures. In R. de Borst & E. Ramm (Eds.), *Multiscale methods in computational mechanics: Progress and accomplishments* (pp. 215–231). Springer Netherlands. (Cit. on p. 2).
- Flaschel, M., Kumar, S., & De Lorenzis, L. (2021).** Unsupervised discovery of interpretable hyperelastic constitutive laws. *Computer Methods in Applied Mechanics and Engineering*, 381, 113852 (cit. on pp. 5, 132, 135).
- Florentin, E., Gallimard, L., & Pelle, J.-P. (2002).** Evaluation of the local quality of stresses in 3d finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 191, 4441–4457 (cit. on p. 9).
- Friedman, J. H. (2001).** Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232 (cit. on p. 5).
- Froggatt, M., & Moore, J. (1998).** High-spatial-resolution distributed strain measurement in optical fiber with rayleigh scatter. *Applied Optics*, 37(10), 1735–1740 (cit. on p. 2).
- Fuhg, J. N., Hamel, C. M., Johnson, K., Jones, R., & Bouklas, N. (2023).** Modular machine learning-based elastoplasticity: Generalization in the context of limited data. *Computer Methods in Applied Mechanics and Engineering*, 407, 115930 (cit. on pp. 31–33, 70).

- Fuhg, J. N., Jones, R. E., & Bouklas, N. (2024).** Extreme sparsification of physics-augmented neural networks for interpretable model discovery in mechanics. *Computer Methods in Applied Mechanics and Engineering*, 426, 116973 (cit. on p. 122).
- Fuhg, J., Bouklas, N., & Jones, R. (2022).** Learning hyperelastic anisotropy from data via a tensor basis neural network. *Journal of the Mechanics and Physics of Solids*, 168, 105022 (cit. on pp. 31, 33).
- Gelb, A. (Ed.). (1974).** *Applied optimal estimation*. M.I.T. Press. (Cit. on p. 115).
- Ghaboussi, J., Wu, X., & Kaklauskas, G. (1999).** Neural network material modelling. *Statyba*, 5(4), 250–257 (cit. on pp. 5, 30).
- Ghosh, S., Zou, Z., Babaniyi, O., Aquino, W., Diaz, M., Bayat, M., & Fatemi, M. (2017).** Modified error in constitutive equations (mece) approach for ultrasound elastography. *Journal of the Acoustical Society of America*, 142(4), 2084 (cit. on p. 15).
- Glisic, B., & Inaudi, D. (2007).** *Fiber optic methods for structural health monitoring*. John Wiley & Sons. (Cit. on p. 2).
- Glorot, X., & Bengio, Y. (2010).** Understanding the difficulty of training deep feedforward neural networks (Y. W. Teh & M. Titterton, Eds.). 9, 249–256 (cit. on pp. 45, 46, 70, 92).
- Goodfellow, I. J., Bengio, Y., & Courville, A. (2016).** *Deep learning*. MIT Press. (Cit. on pp. 5, 22, 25, 49, 82).
- Gorji, M. B., Mozaffar, M., Heidenreich, J. N., Cao, J., & Mohr, D. (2020).** On the potential of recurrent neural networks for modeling path dependent plasticity. *Journal of The Mechanics and Physics of Solids*, 143, 103972 (cit. on pp. 34, 90).
- Gove, J. H., & Hollinger, D. Y. (2006).** Application of a dual unscented Kalman filter for simultaneous state and parameter estimation in problems of surface-atmosphere exchange. *Journal of Geophysical Research: Atmospheres*, 111(D8), 2005JD006021 (cit. on p. 111).
- Grédiac, M. (2004).** The use of full-field measurement methods in composite material characterization: Interest and limitations. *Composites Part A: Applied Science and Manufacturing*, 35(7), 751–761 (cit. on pp. 5, 35).
- Guchhait, S., & Banerjee, B. (2015).** Constitutive error based material parameter estimation procedure for hyperelastic material. *Computer Methods in Applied Mechanics and Engineering*, 297, 455–475 (cit. on p. 15).
- Guchhait, S., & Banerjee, B. (2016).** Anisotropic linear elastic parameter estimation using error in the constitutive equation functional. *Proceedings of the Royal Society A*, 472, 20160213 (cit. on p. 15).

- Guchhait, S., & Banerjee, B. (2018).** Constitutive error based parameter estimation technique for plate structures using free vibration signatures. *Journal of Sound and Vibration*, 419, 302–317 (cit. on p. 15).
- Gulgec, N. S., Takáč, M., & Pakzad, S. N. (2017).** Structural damage detection using convolutional neural networks. In R. Barthorpe, R. Platz, I. Lopez, B. Moaveni, & C. Papadimitriou (Eds.), *Model validation and uncertainty quantification, volume 3*. Springer. (Cit. on p. 31).
- Guzman De Vioria, R., Yamamoto, N., Miravete, A., & Wardle, B. L. (2011).** Multi-physics damage sensing in nano-engineered structural composites. *Nanotechnology*, 22, 185502 (cit. on p. 2).
- Haasdonk, B., Kleikamp, H., Ohlberger, M., Schindler, F., & Wenzel, T. (2023).** A new certified hierarchical and adaptive rb-ml-rom surrogate model for parametrized pdes. *SIAM Journal on Scientific Computing*, 45(3), A1039–A1065 (cit. on p. 126).
- Hadj-Sassi, K. (2007).** *Une stratégie d'estimation conjointe des paramètres et de l'état de structures à comportements non linéaires. assimilation de données et erreur en loi de comportement* (Doctoral dissertation). Ecole Polytechnique. Paris. (Cit. on p. 15).
- Haghighat, E., Raissi, M., Moure, A., Gomez, H., & Juanes, R. (2021).** A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379, 113741 (cit. on p. 28).
- Halphen, B., & Nguyen, Q. (1975).** On generalized standard materials. *Journal de Mécanique*, 14, 39–63 (cit. on pp. 5, 16, 18, 32).
- He, K., Zhang, X., Ren, S., & Sun, J. (2015).** Delving deep into rectifiers: Surpassing human-level performance on imagenet classification In *2015 IEEE international conference on computer vision (iccv)*. 1026–1034. (Cit. on p. 46).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016).** Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (cit. on p. 21).
- He, X., & Chen, J.-S. (2022).** Thermodynamically consistent machine-learned internal state variable approach for data-driven modeling of path-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 402, 115348 (cit. on p. 32).
- Heidenreich, J. N., Bonatti, C., & Mohr, D. (2024).** Transfer learning of recurrent neural network-based plasticity models. *International Journal for Numerical Methods in Engineering*, 125(1), e7357 (cit. on p. 29).
- Hénault, J., Quiertant, M., Delepine-Lesoille, S., Salin, J., Moreau, G., Taillade, F., & Benzarti, K. (2012).** Quantitative strain measurement and crack detection in rc structures using a truly distributed fiber optic sensing system. *Construction and Building Materials*, 37, 916–923 (cit. on p. 2).

- Hernández, Q., Badías, A., González, D., Chinesta, F., & Cueto, E. (2021).** Structure-preserving neural networks. *Journal of Computational Physics*, 426, 109950 (cit. on p. 32).
- Hild, F., & Roux, S. (2006).** Digital image correlation: From displacement measurement to identification of elastic properties—a review. *Strain*, 42(2), 69–80 (cit. on p. 2).
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012).** Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups In *Ieee signal processing magazine*. IEEE. 82–97. (Cit. on p. 21).
- Hinton, G. E., & Salakhutdinov, R. R. (2006).** Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507 (cit. on p. 24).
- Hochreiter, S., & Schmidhuber, J. (1997).** Long short-term memory. *Neural computation*, 9(8), 1735–1780 (cit. on p. 24).
- Horace He, R. Z. (2021).** Functorch: Jax-like composable function transforms for pytorch. *Library available on GitHub* (cit. on p. 51).
- Hornik, K. (1991).** Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251–257 (cit. on p. 24).
- Hu, X., Chodora, E., Prabhu, S., & Atamturktur, S. (2019).** Extended constitutive relation error-based approach: The role of mass in damage detection. *Structural Control and Health Monitoring*, 26(5), e2318 (cit. on p. 15).
- Hu, X., Prabhu, S., Atamturktur, S., & Cogan, S. (2017).** Mechanistically-informed damage detection using dynamic measurements: Extended constitutive relation error. *Mechanical Systems and Signal Processing*, 85, 312–328 (cit. on p. 15).
- Huang, S., Feissel, P., & Villon, P. (2016).** Modified constitutive relation error: An identification framework dealing with the reliability of information. *Computer Methods in Applied Mechanics and Engineering*, 311, 1–17 (cit. on p. 15).
- Ibragimova, O., Brahme, A., Muhammad, W., Lévesque, J., & Inal, K. (2021).** A new ann based crystal plasticity model for fcc materials and its application to non-monotonic strain paths. *International Journal of Plasticity*, 144, 103059 (cit. on p. 34).
- Im, S., Lee, J., & Cho, M. (2021).** Surrogate modeling of elasto-plastic problems via long short-term memory neural networks and proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 385, 114030 (cit. on p. 34).
- Intergovernmental Panel on Climate Change. (2019).** Climate change and land: An ipcc special report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems. (Cit. on p. 1).

- Jacobs, R. A. (1988).** Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4), 295–307 (cit. on p. 47).
- Jordan, B., Gorji, M. B., & Mohr, D. (2020).** Neural network model describing the temperature- and rate-dependent stress-strain response of polypropylene. *International Journal of Plasticity*, 135, 102811 (cit. on p. 135).
- Julier, S., Uhlmann, J., & Durrant-Whyte, H. (2000).** A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3), 477–482 (cit. on p. 110).
- Julier, S. J., & Uhlmann, J. K. (1997).** New extension of the Kalman filter to nonlinear systems In *Signal processing, sensor fusion, and target recognition vi* (I. Kadar, Ed.). International Society for Optics and Photonics. SPIE. 182–193. (Cit. on pp. 110, 115, 116).
- Julier, S., Uhlmann, J., & Durrant-Whyte, H. (1995).** A new approach for filtering nonlinear systems In *Proceedings of 1995 american control conference - acc'95*. 1628–1632 vol.3. (Cit. on p. 110).
- Kaipio, J., & Somersalo, E. (2007).** Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198, 493–504 (cit. on pp. 5, 35).
- Kalman, R. E. (1960).** A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1), 35–45 (cit. on pp. 110, 113).
- Kingma, D. P., & Ba, J. (2015).** Adam: A method for stochastic optimization (Y. Bengio & Y. LeCun, Eds.) (cit. on pp. 26, 46, 76).
- Kohn, B. D., Robert V Lowe. (1988).** A variational method for parameter identification. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 22(1), 119–158 (cit. on p. 10).
- Kovacs, A., Exl, L., Kornell, A., Fischbacher, J., Hovorka, M., Gusenbauer, M., Breth, L., Oezelt, H., Praetorius, D., Suess, D., & Schrefl, T. (2022).** Magnetostatics and micromagnetics with physics informed neural networks. *Journal of Magnetism and Magnetic Materials*, 548, 168951 (cit. on p. 28).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017).** Imagenet classification with deep convolutional neural networks In *Communications of the acm*. ACM. 84–90. (Cit. on p. 21).
- Ladevèze, P. (1975).** *Comparaison de modèles de milieux continus* (Doctoral dissertation). PhD thesis. (Cit. on p. 8).

- Ladevèze, P. (2001).** Constitutive relation errors for f.e. analysis considering (visco-) plasticity and damage. *International Journal for Numerical Methods in Engineering*, 52(5-6), 527–542 (cit. on p. 18).
- Ladevèze, P., Blaysat, B., & Florentin, E. (2012).** Strict upper bounds of the error in calculated outputs of interest for plasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 245-246, 194–205 (cit. on p. 19).
- Ladevèze, P., & Chamoin, L. (2010).** Calculation of strict error bounds for finite element approximations of nonlinear pointwise quantities of interest. *International Journal for Numerical Methods in Engineering*, 84, 1638–1664 (cit. on p. 19).
- Ladevèze, P., & Maunder, E. (1996).** A general method for recovering equilibrating element tractions. *Computer Methods in Applied Mechanics and Engineering*, 137, 111–151 (cit. on p. 9).
- Ladevèze, P., & Moës, N. (1998).** A new a posteriori error estimation for nonlinear time-dependent finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 157(1), 45–68 (cit. on p. 18).
- Ladevèze, P., Puel, G., Deraemaeker, A., & Romeuf, T. (2006).** Validation of structural dynamics models containing uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 195(4), 373–393 (cit. on p. 15).
- Ladevèze, P., & Reynier, M. (1989).** A localization method of stiffness errors and adjustments of fe models, 355–361 (cit. on pp. 10, 133).
- Ladevèze, P., Reynier, M., & Maia, N. (1994).** Error on the constitutive relation in dynamics: Theory and application for model updating. In H. D. Bui & M. Tanaka (Eds.), *Inverse problems in engineering* (pp. 251–256). (Cit. on pp. 5, 10, 15).
- Ladevèze, P., & Rougeot, P. (1997).** New advances on a posteriori error on constitutive relation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 150, 239–249 (cit. on p. 8).
- Ladevèze, P. (1999).** Principles of the method of large time increments. In *Nonlinear computational structural mechanics: New approaches and non-incremental methods of calculation* (pp. 55–100). Springer New York. (Cit. on pp. 20, 66, 126).
- Ladevèze, P., Allix, O., & Daudeville, L. (1991).** Mesomodeling of damage for laminate composites: Application to delamination In *Inelastic deformation of composite materials: Iutam symposium, troy, new york, may 29–june 1, 1990*. Springer New York. 607–622. (Cit. on p. 2).
- Ladevèze, P., & Chamoin, L. (2016).** The constitutive relation error method: A general verification tool. In L. Chamoin & P. Díez (Eds.), *Verifying calculations - forty years on: An overview of classical verification techniques for fem simulations* (pp. 59–94). Springer International Publishing. (Cit. on p. 9).

- Ladevèze, P., & Chouaki, A. (1999).** Application of a posteriori error estimation for structural model updating. *Inverse Problems*, 15, 49 (cit. on p. 15).
- Ladevèze, P., & Leguillon, D. (1983).** Error estimate procedure in the finite element method and applications. *Siam Journal on Numerical Analysis*, 20, 485–509 (cit. on pp. 8, 9).
- Ladevèze, P., & Moës, N. (1997).** A new a posteriori error estimation for nonlinear time-dependent finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 157, 45–68 (cit. on p. 19).
- Ladevèze, P., & Pelle, J.-P. (2004).** Mastering calculation in linear and nonlinear mechanics (cit. on pp. 9, 18).
- Lagaris, I. E., Likas, A. C., & Fotiadis, D. I. (1997).** Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9, 987–1000 (cit. on p. 27).
- Lagergren, J. H., Nardini, J. T., Baker, R. E., Simpson, M. J., & Flores, K. B. (2020).** Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. *PLOS Computational Biology*, 16(12), 1–29 (cit. on p. 27).
- Law, K., Stuart, A., & Zygalakis, K. (2015).** *Data Assimilation: A Mathematical Introduction* (Vol. 62). Springer International Publishing. (Cit. on p. 110).
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998).** Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324 (cit. on p. 24).
- Lefebvre, T., Bruyninckx, H., & De Schutter, J. (2004).** Kalman filters for non-linear systems: A comparison of performance. *International Journal of Control*, 77(7), 639–653 (cit. on p. 115).
- Lemaitre, J., & Chaboche, J.-L. (1990).** *Mechanics of solid materials*. Cambridge University Press. (Cit. on pp. 2, 5, 16, 83).
- Li, L., & Chen, C. Q. (2022).** Equilibrium-based convolution neural networks for constitutive modeling of hyperelastic materials. *Journal of the Mechanics and Physics of Solids*, 164, 104931 (cit. on p. 135).
- Li, W., Sun, S., Jia, Y., & Du, J. (2016).** Robust unscented Kalman filter with adaptation of process and measurement noise covariances. *Digital Signal Processing*, 48, 93–103 (cit. on p. 111).
- Lim, S. Y., & Lim, K. H. (2022).** Second-order derivative optimization methods in deep learning neural networks In *2022 international conference on green energy, computing and sustainable technology (gecost)*. 470–475. (Cit. on p. 102).
- Linden, L., Klein, D. K., Kalina, K. A., Brummund, J., Weeger, O., & Kästner, M. (2023).** Neural networks meet hyperelasticity: A guide to enforcing physics. *Journal of the Mechanics and Physics of Solids*, 179, 105363 (cit. on pp. 31, 33, 70).

- Linka, K., & Kuhl, E. (2023).** A new family of constitutive artificial neural networks towards automated model discovery. *Computational Methods in Applied Mechanics and Engineering*, 403, 115731 (cit. on pp. 31, 33, 34).
- Liu, D., Yang, H., Elkhodary, K., Tang, S., Liu, W. K., & Guo, X. (2022).** Mechanistically informed data-driven modeling of cyclic plasticity via artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 393, 114766 (cit. on p. 31).
- Liu, P., & Zheng, J. (2010).** Recent developments on damage modeling and finite element analysis for composite laminates: A review. *Materials & Design*, 31(8), 3825–3834 (cit. on p. 2).
- Maday, Y., Patera, A., Penn, J. D., & Yano, M. (2015).** A parameterized-background data-weak approach to variational data assimilation: Formulation, analysis, and application to acoustics. *International Journal in Numerical Methods in Engineering*, 102(5), 933–965 (cit. on p. 3).
- Marchand, B. (2017).** *Assimilation de données et recalage rapide de modèles mécaniques complexes* (Theses 2017SACLN053). Université Paris Saclay (COMUE). (Cit. on p. 110).
- Marchand, B., Chamoin, L., & Rey, C. (2016).** Real-time updating of structural mechanics models using kalman filtering, modified constitutive relation error, and proper generalized decomposition. *International Journal for Numerical Methods in Engineering*, 107(9), 786–810 (cit. on pp. 90, 111, 118, 119, 126).
- Marchand, B., Chamoin, L., & Rey, C. (2019).** Parameter identification and model updating in the context of nonlinear mechanical behaviors using a unified formulation of the modified constitutive relation error concept. *Computer Methods in Applied Mechanics and Engineering*, 345, 1094–1113 (cit. on pp. 15, 20, 64, 66).
- Mariani, S., & Corigliano, A. (2005).** Impact induced composite delamination: State and parameter identification via joint and dual extended Kalman filters. *Computer Methods in Applied Mechanics and Engineering*, 194(50-52), 5242–5272 (cit. on pp. 111, 118).
- Mariani, S., & Ghisi, A. (2007).** Unscented Kalman filtering for nonlinear structural dynamics. *Nonlinear Dynamics*, 49(1-2), 131–150 (cit. on pp. 111, 115).
- Masi, F., & Stefanou, I. (2022).** Multiscale modeling of inelastic materials with thermodynamics-based artificial neural networks (tann). *Computer Methods in Applied Mechanics and Engineering*, 398, 115190 (cit. on pp. 32, 34).
- Massala, S., Chamoin, L., & Pica Ciamarra, M. (2023).** Hybrid twins coupling PBDW and machine learning for effective state estimation on complex systems In *2nd IACM Conference on Mechanics Machine Learning and Digital Engineering for Computational Science Engineering and Technology (MMLDE-CSET)*. El Paso, United States. (Cit. on p. 29).

- McClellan, A., Lorenzetti, J., Pavone, M., & Farhat, C. (2022).** A physics-based digital twin for model predictive control of autonomous unmanned aerial vehicle landing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2229), 20210204 (cit. on p. 3).
- McCulloch, W. S., & Pitts, W. (1943).** A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133 (cit. on p. 22).
- Mielke, A. (2011).** Formulation of thermoelastic dissipative material behavior using generic. *Continuum Mechanics and Thermodynamics*, 23, 233–256 (cit. on p. 32).
- Mooney, M. (1940).** A Theory of Large Elastic Deformation. *Journal of Applied Physics*, 11(9), 582–592 (cit. on p. 55).
- Morozov, V. (1968).** The error principle in the solution of operational equations by the regularization method. *USSR Computational Mathematics and Mathematical Physics*, 8(2), 63–87 (cit. on pp. 13, 47).
- Mozaffar, M., Bostanabad, R., Bostanabad, R., Chen, W., Ehmman, K. F., Cao, J., & Bessa, M. A. (2019).** Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences of the United States of America*, 116, 26414–26420 (cit. on p. 34).
- Muralidhar, N., Bu, J., Cao, Z., He, L., Ramakrishnan, N., Tafti, D., & Karpatne, A. (2020).** Phynet: Physics guided neural networks for particle drag force prediction in assembly. In *Proceedings of the 2020 siam international conference on data mining (sdm)* (pp. 559–567). (Cit. on p. 29).
- Nguyen, H. N. (2021).** *New numerical strategies for robust, consistent, and computationally efficient model identification from full-field measurements* (Theses 2021UPAST061). Université Paris-Saclay. (Cit. on pp. 12, 15, 20, 46, 65, 66, 79, 92).
- Nguyen, H., Allix, O., & Feissel, P. (2008).** A robust identification strategy for rate-dependent models in dynamics. *Inverse Problems*, 24, 065006 (cit. on pp. 15, 16, 111).
- Nocedal, J., & Wright, S. J. (2006).** *Numerical optimization*. Springer. (Cit. on p. 68).
- Oden, J., Belytschko, T., Fish, J., Hughes, T., Johnson, C., Keyes, D., Laub, A., Petzold, L., Srolovitz, D., & Yip, S. (2006).** *Simulation-based engineering science: Revolutionizing engineering science through simulation*. NSF Blue Ribbon Panel on SBES. (Cit. on p. 3).
- Onat, A. (2019).** A novel and computationally efficient joint unscented kalman filtering scheme for parameter estimation of a class of nonlinear systems. *IEEE Access*, 7, 31634–31655 (cit. on p. 111).
- Öttinger, H. C., & Grmela, M. (1997).** Dynamics and thermodynamics of complex fluids. ii. illustrations of a general formalism. *Phys. Rev. E*, 56, 6633–6655 (cit. on pp. 5, 32).

- Papakonstantinou, K. G., Amir, M., & Warn, G. P. (2022).** A Scaled Spherical Simplex Filter (S3F) with a decreased $n + 2$ sigma points set size and equivalent $2n + 1$ Unscented Kalman Filter (UKF) accuracy. *Mechanical Systems and Signal Processing*, 163, 107433 (cit. on pp. 110, 115).
- Pares, N., Diez, P., & Huerta, A. (2006).** Subdomain-based flux-free a posteriori error estimators. *Computer Methods in Applied Mechanics and Engineering*, 195, 297–323 (cit. on p. 10).
- Pares, N., Diez, P., & Huerta, A. (2009).** Exact bounds for linear outputs of the advection-diffusion-reaction equation using flux-free error estimates. *SIAM Journal on Scientific Computing*, 31(4), 3064–3089 (cit. on p. 10).
- Peherstorfer, B., & Willcox, K. (2015a).** Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4), A2123–A2150 (cit. on p. 126).
- Peherstorfer, B., & Willcox, K. (2015b).** Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291, 21–41 (cit. on p. 3).
- Pierre, S. R. S., Rajasekharan, D., Darwin, E. C., Linka, K., Levenston, M. E., & Kuhl, E. (2023).** Discovering the mechanics of artificial and real meat. *Computer Methods in Applied Mechanics and Engineering*, 415, 116236 (cit. on p. 135).
- Pipard, J.-M., Balan, T., Abed-Meraim, F., & Lemoine, X. (2013).** Physically-motivated elasto-visco-plastic model for the large strain-rate behavior of steels. *Key Engineering Materials*, 554-557, 1164–1173 (cit. on p. 83).
- Pled, F., Chamoin, L., & Ladevèze, P. (2011).** On the techniques for constructing admissible stress fields in model verification: Performances on engineering examples. *International Journal for Numerical Methods in Engineering*, 88(5), 409–441 (cit. on p. 10).
- Quarteroni, A., Rozza, G., & Manzoni, A. (2011).** Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(3), 1–49 (cit. on p. 2).
- Raissi, M., Perdikaris, P., & Karniadakis, G. (2019).** Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707 (cit. on pp. 5, 26, 27).
- Rao, C., & Liu, Y. (2020).** Three-dimensional convolutional neural network (3d-cnn) for heterogeneous material homogenization. *Computational Materials Science*, 184, 109850 (cit. on p. 31).

- Rausch, J., & Mäder, E. (2010).** Health monitoring in continuous glass fibre reinforced thermoplastics: Tailored sensitivity and cyclic loading of cnt-based interphase sensors. *Composites Science and Technology*, 70, 2023–2030 (cit. on p. 2).
- Relun, N., Néron, D., & Boucard, P. (2013).** A model reduction technique based on the pgd for elastic-viscoplastic computational analysis. *Computational Mechanics*, 51, 83–92 (cit. on p. 126).
- Rezaei, S., Moeineddin, A., & Rajaei Harandi, A. (2024).** Learning solutions of thermodynamics-based nonlinear constitutive material models using physics-informed neural networks. *Computational Mechanics*, 1, 1–34 (cit. on p. 30).
- Richardson, L. F., & Lynch, P. (1922).** *Weather Prediction by Numerical Process* (2nd ed.). Cambridge University Press, Second edition (2007), original paper in 1922. (Cit. on p. 110).
- Rivlin, R. S. (1948).** Large elastic deformations of isotropic materials. iv. further developments of the general theory. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 241(835), 379–397 (cit. on p. 55).
- Robbins, H., & Monro, S. (1951).** A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3), 400–407 (cit. on p. 25).
- Romero, I. (2009).** Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. *International Journal for Numerical Methods in Engineering*, 79, 706–732 (cit. on p. 32).
- Rosenblatt, F. (1958).** The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 386–408 (cit. on p. 22).
- Rosenkranz, M., Kalina, K. A., Brummund, J., Sun, W., & Kästner, M. (2024).** Viscoelasticity with physics-augmented neural networks: Model formulation and training methods without prescribed internal variables. *arXiv preprint arXiv:2401.14270* (cit. on pp. 32, 34, 126).
- Rosic, B., Kucerova, A., Sykora, J., Pajonk, O., Litvinenko, A., & Matthies, H. (2013).** Parameter identification in a probabilistic setting. *Engineering Structures*, 50, 179–196 (cit. on pp. 5, 35).
- Rytter, A. (1993).** *Vibrational based inspection of civil engineering structures* (Doctoral dissertation). Department of Building Technology and Structural Engineering, Aalborg University. (Cit. on p. 1).
- Saxe, A., McClelland, J., & Ganguli, S. (2014).** Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. 1–22. (Cit. on p. 46).

- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008).** The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 61–80 (cit. on p. 25).
- Schwartz, M. (2008).** *Encyclopedia of smart materials* (M. Schwartz, Ed.). CRC Press. (Cit. on p. 2).
- Si, X., Wang, W., Hu, C., & Zhou, D. (2011).** Remaining useful life estimation – a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1), 1–14 (cit. on p. 1).
- Silva, T., & Maia, N. (2017).** Detection and localisation of structural damage based on the error in the constitutive relations in dynamics. *Applied Mathematical Modelling*, 46, 736–749 (cit. on p. 15).
- Smith, L. N. (2017).** Cyclical learning rates for training neural networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)* (cit. on p. 53).
- Soller, B., Gifford, D., Wolfe, M., & Frogatt, M. (2005).** High resolution optical frequency domain reflectometry characterization of components and assemblies. *Optics Express*, 13, 666–674 (cit. on p. 2).
- Sunahara, Y., & Yamashita, K. (1970).** An approximate method of state estimation for non-linear dynamical systems with state-dependent noise†. *International Journal of Control*, 11(6), 957–972 (cit. on p. 110).
- Sutton, M. A., Wolters, W., Peters, W., Ranson, W., & McNeill, S. (1983).** Determination of displacements using an improved digital correlation method. *Image and vision computing*, 1(3), 133–139 (cit. on p. 2).
- Tac, V., Sahli Costabal, F., & Tepole, A. (2022).** Data-driven tissue mechanics with poly-convex neural ordinary differential equations. *Computational Methods in Applied Mechanics and Engineering*, 398, 115248 (cit. on pp. 31, 33).
- Tarantola, A. (2005).** *Inverse problem theory and model parameter estimation*. SIAM. (Cit. on pp. 5, 35).
- Thakolkaran, P., Joshi, A., Zheng, Y., Flaschel, M., De Lorenzis, L., & Kumar, S. (2022).** Nn-euclid: Deep-learning hyperelasticity without stress data. *Journal of the Mechanics and Physics of Solids*, 169, 105076 (cit. on pp. 31, 33, 35, 46, 52, 82, 94, 131, 132).
- Thostenson, E. T., & Chou, T.-W. (2006).** Carbon nanotube networks: Sensing of distributed strain and damage for life prediction and self-healing. *Advanced Materials*, 18, 2837–2841 (cit. on p. 2).
- Tieleman, T., & Hinton, G. (2012).** Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural networks for machine learning*, 4(2) (cit. on p. 26).

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is all you need In *Advances in neural information processing systems*. 5998–6008. (Cit. on p. 21).
- Vlassis, N. N., & Sun, W. (2021).** Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Computer Methods in Applied Mechanics and Engineering*, 377, 113695 (cit. on p. 34).
- Waeysens, J., Chamoin, L., & Ladevèze, P. (2012).** Guaranteed error bounds on point-wise quantities of interest for transient viscodynamics problems. *Computational Mechanics*, 49(3), 291–307 (cit. on p. 19).
- Waeysens, J., Rosić, B., Charbonnel, P.-E., Merliot, E., Siegert, D., Chapeleau, X., Vidal, R., Corvec, V. I., & Cottineau, L.-M. (2016).** Model updating techniques for damage detection in concrete beam using optical fiber strain measurement device. *Engineering Structures*, 129, 2–10 (cit. on pp. 5, 15).
- Wang, G., & Ke, J. (2024).** Literature review on the structural health monitoring (shm) of sustainable civil infrastructure: An analysis of influencing factors in the implementation. *Buildings*, 14(2), 402 (cit. on p. 1).
- Wang, S., Teng, Y., & Perdikaris, P. (2020).** Understanding and mitigating gradient pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.*, 43, A3055–A3081 (cit. on p. 46).
- Warner, J., Diaz, M., Aquino, W., & Bonnet, M. (2014).** Inverse material identification in coupled acoustic-structure interaction using a modified error in constitutive equation functional. *Computational Mechanics*, 54, 645–659 (cit. on p. 15).
- Weber, P., Wagner, W., & Freitag, S. (2023).** Physically enhanced training for modeling rate-independent plasticity with feedforward neural networks. *Computational Mechanics*, 72, 827–857 (cit. on p. 34).
- Wu, L., Nguyen, V. D., Kilingar, N. G., & Noels, L. (2020).** A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. *Computer Methods in Applied Mechanics and Engineering* (cit. on p. 34).
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016).** Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (cit. on p. 21).
- Zanardi, I., Venturi, S., & Panesi, M. (2022).** Towards efficient simulations of non-equilibrium chemistry in hypersonic flows: A physics-informed neural network framework In *Aiaa scitech 2022 forum*. (Cit. on p. 28).

Zhang, A., & Mohr, D. (2020). Using neural networks to represent von mises plasticity with isotropic hardening. *International Journal of Plasticity*, 369, 113234 (cit. on p. 34).