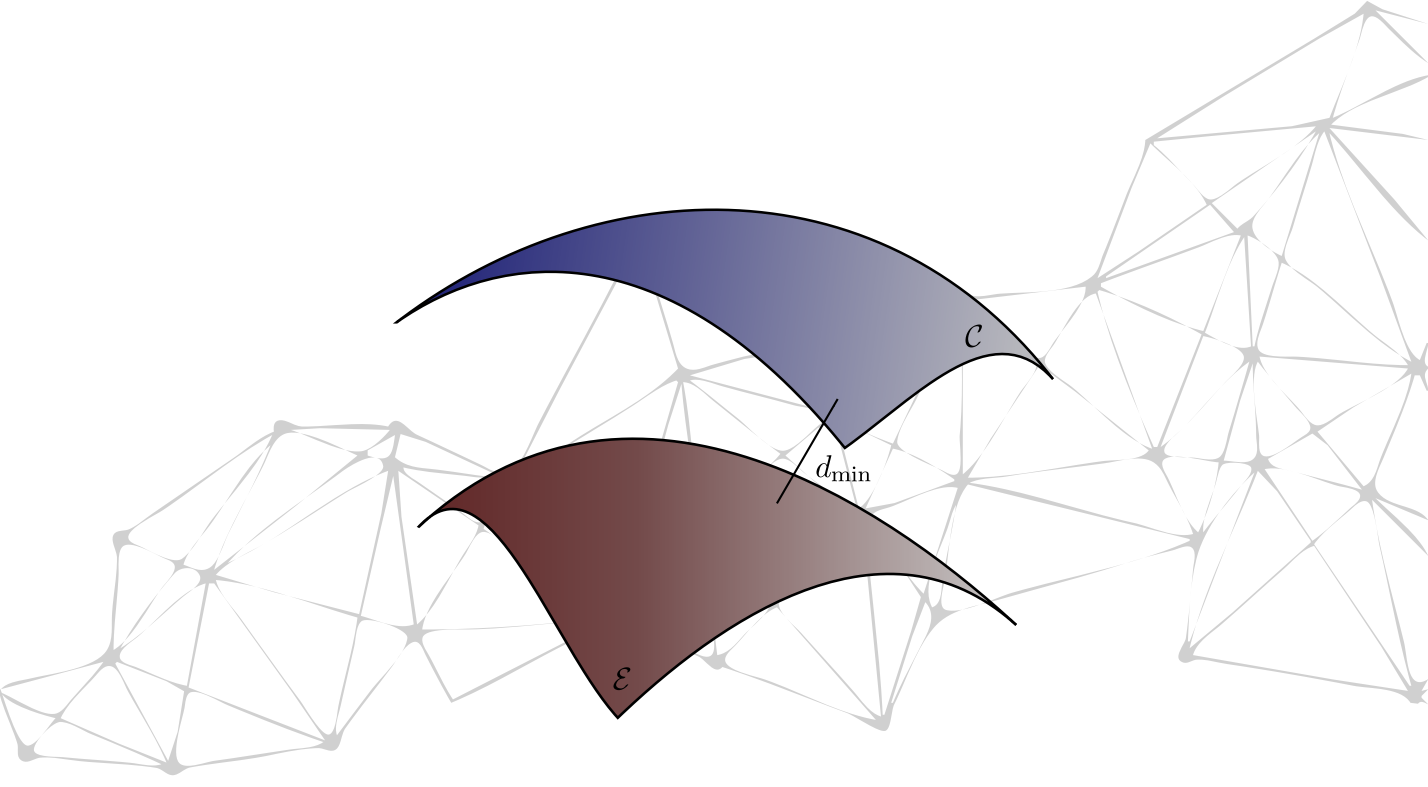
RUHR-UNIVERSITÄT BOCHUM

RUB

An Investigation of Advances for Model-Free Data-Driven Computational Mechanics:

Tangent Space Information, Haigh-Westergaard Space, and Neural Networks

Kerem Ciftci



RUHR-UNIVERSITÄT BOCHUM
Institut für Mechanik

An Investigation of Advances for Model-Free Data-Driven Computational Mechanics: Tangent Space Information, Haigh-Westergaard Space, and Neural Networks

Zur Erlangung des akademischen Grades

Dr.-Ing.

vorgelegt der

Fakultät für Bau- und Umweltingenieurwissenschaften

an der Ruhr-Universität Bochum

von

Kerem Ciftci
geb. 03.06.1990
in Siegen

Mitteilungen aus dem Institut für Mechanik Nr. 195

Herausgeber (Publisher):

Institut für Mechanik
— Schriftenreihe —
Ruhr-Universität Bochum
D-44780 Bochum

ISBN 978-3-935892-73-5

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by the copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by the author's copyright. These works or parts of it may not be used to repost reprint/republish or for creating new collective works for resale or redistribution to servers or lists without the explicit permission of the copyright holder.

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

©2024 Kerem Ciftci, Lehrstuhl für Mechanik - Materialtheorie der Ruhr-Universität Bochum

Printed in Germany

Einreichung der Dissertation (thesis submission):	10.04.2024
Mündliche Prüfung (thesis defense):	05.09.2024

Erster Gutachter (first referee):	Prof. Dr. rer. nat. Klaus Hackl
Zweiter Gutachter (second referee):	Prof.'in Dr.-Ing. Johanna Waimann
Dritter Gutachter (third referee):	Prof. Dr.-Ing. Marc-André Keip
Vorsitzender (committee chair):	Prof. Dr.-Ing. Iurie Curoşu

Dedicated to my family, wife and friends.

Acknowledgments

I express my deepest gratitude to Prof. Dr. Hackl, my thesis supervisor, for his unwavering support and guidance throughout this research. Your expertise and insights have been invaluable to my work.

A special thanks to my colleagues and peers at Ruhr University Bochum, particularly Dr. Hoppe and Dr. Riedel, for their collaboration, encouragement, and engaging discussions that enriched my research experience.

I am especially thankful to my family, particularly my supportive mother, Gülseren, whose strength, resilience, and endless dedication to our family have been a constant source of inspiration. Her sacrifices and all she has endured have built the foundation of my life, and for that, I am forever grateful.
I also want to thank my loving wife, Madina, for her incredible support throughout my academic journey. Her love, patience, and constant encouragement have been my source of strength. I am truly grateful for everything she has done to help me, and I could not have made it this far without her by my side.
Not to forget, I express my deepest thanks to my brother, Deniz, who has been both a father figure and my closest friend. I am proud of everything he has done for me and our family.

Lastly, I want to thank my friends and everyone who has been part of my journey for their understanding, support, and unwavering faith in me.

This journey would not have been possible without the contributions and support of all these individuals and groups. I am deeply grateful for their involvement in my academic journey.

Gratefully and ready for a long vacation, Kerem Ciftci.

Summary

Computational solid mechanics employs numerical simulations and modeling to analyze and predict the responses of structures under diverse forces, solving intricate physical problems. These simulations, crucial when large-scale experiments are not feasible, use discretization methods such as finite elements and mesh-free approaches to model complex material behaviors. The equations used in solid mechanics, derived from continuum mechanical theory, comprise three critical components:

1. Ensuring the balance of linear momentum.
2. Confirming the compatibility of kinematic quantities.
3. Applying constitutive mappings that define the relationship between stress and strain.

While the first two conditions are based on fundamental physics and geometry, the constitutive model relies on data and engineering knowledge. Traditionally, this involved creating models based on material experiments. Material modeling approaches can be broadly classified into phenomenological modeling, which focuses on capturing a material's macroscopic behavior, and microscopic modeling, which aims to predict material behavior by simulating the underlying physical processes at the atomic or microscopic level. Modern modeling utilizes machine learning and data-driven techniques to enhance predictions and facilitate the development of new materials.

A fundamental challenge in traditional material modeling arises when there is no direct intersection between the material model solutions and the constraints of physical laws. Kirchdoerfer and Ortiz introduced a relaxation approach called *model-free data-driven computational mechanics*, which directly utilizes experimental datasets to solve mechanical problems, bypassing traditional material modeling to resolve this issue. The methodological innovation lies in determining the state within the constraints closest to the material data set. The state-of-the-art in this domain has witnessed significant growth in areas such as elasticity, finite strain, and damage mechanics. The approach is particularly notable in its ability to derive solutions that adhere more closely to the observed data. Machine learning further extends the capabilities of model-free methods, offering solutions to the difficulties presented by noisy, sparse, or extensive datasets.

Disseration's outline: The cumulative dissertation advances model-free data-driven computational mechanics by integrating history-dependent material properties and optimizing data usage. The first article introduces a hybrid approach that enriches the data set with tangent space directions to capture directional changes in material behavior, tailored for materials with complex loading histories like non-linear elasticity and elasto-plasticity. The second article optimizes the data collection process using material symmetry and Haigh-Westergaard coordinates, enhancing the accuracy of simulations for materials under elasto-plastic deformation with isotropic hardening. The third article suggests leveraging a Generative Adversarial Network framework to enrich the data-driven methodology. It describes how physical constraints inform the generator while the discriminator assesses its output against the nearest strain-stress data points. This approach combines data-driven mechanics with deep learning, indicating a move towards sophisticated machine learning applications to improve predictive simulations.

Declaration

I declare on oath that I have written the submitted doctoral thesis independently and without unauthorised third-party assistance, have not used any literature other than the literature stated, and that I have indicated all sections of text incorporated in full or paraphrased as well as any graphics, tables and evaluation programmes used.

Furthermore, I affirm that the submitted electronic version corresponds to the written version of the doctoral thesis and the paper has not been submitted or assessed as doctoral work in this or a similar form.

Published Content and Contributions

1. CIFTCI, K., & HACKL, K. (2022). Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules. *Computational Mechanics, 70(2)*, 425-435.

   The first author conceptualized and developed all computational code and methodologies, derived the theoretical proofs, and composed the manuscript. The second author conceived the main idea, contributed to the conceptual framework, provided critical feedback, and helped shape the research, analysis, and manuscript.

2. CIFTCI, K., & HACKL, K. (2023). Model-free data-driven inelasticity in Haigh-Westergaard space – A study on how to obtain data points from measurements. *Computer Methods in Applied Mechanics and Engineering, 416*, 116352.

   The first author conceptualized and developed all computational code and methodologies, derived the theoretical proofs, and composed the manuscript. The second author conceived the main idea, contributed to the conceptual framework, provided critical feedback, and helped shape the research, analysis, and manuscript.

3. CIFTCI, K., & HACKL, K. (2024). A physics-informed GAN framework based on model-free data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering, 424*, 116907.

   The first author derived the main idea, was responsible for the conceptualization and development of all computational code and methodologies, derived the theoretical proofs, and was responsible for the composition of the manuscript. The second author contributed to the conceptual framework, provided critical feedback, and helped shape the research, analysis, and manuscript.

Contents

1 Introduction 1

1.1 Computational solid mechanics 1

1.1.1 Perspective on material modeling 4

1.1.2 Integration of machine learning in model-based approaches 5

1.2 Model-free data-driven computational mechanics 6

1.2.1 State-of-the-art 8

1.2.2 Integration of machine learning in model-free approaches 14

1.3 Outline of the enclosed articles 16

2 Article 1: Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules 19

2.1 Introduction 20

2.2 Classical data-driven computing paradigm 21

2.3 Extension by tangent space 23

2.3.1 Data-driven formulation 23

2.3.2 Transition rules 25

2.4 Numerical examples 27

2.4.1 Non-linear elastic cylindrical tube under internal pressure 28

2.4.2 Elasto-plastic plate with a circular hole 31

2.5 Conclusions 34

3 Article 2: Model-free data-driven inelasticity in Haigh-Westergaard space – a study how to obtain data points from measurements 35

3.1 Introduction 36

3.2 Tangent space enhanced data-driven paradigm 38

3.2.1 Tangent space, structured data sets and transition rules 39

3.3 Data-driven paradigm in octahedral plane 41

3.3.1 Preliminaries for tangents of isotropic elasto-plastic bodies 41

3.3.2 Normal vector in the octahedral plane based on Haigh-Westergaard coordinates 42

3.3.3 Data enforced tangent 45

3.3.4 Adapted data-driven projection $P_{\mathcal{D}}$ 46

3.4 Numerical result for a $3D$ benchmark 50

3.4.1 Plate with a circular hole 50

3.5 Conclusion 56

3.A Algorithm of the modified data-driven solver 57

4 Article 3: A physics-informed GAN framework based on model-free data-driven computational mechanics 59

4.1 Introduction 60

4.2 Model-free Data-driven setting 62

4.3 GANs with physics-informed generators for model-free data-driven problems 63

4.3.1 Physics-informed neural networks 63

4.3.2 Intermezzo to generative adversarial networks 66
4.3.3 Physics-informed GANs for data-driven mechanics problems 68
4.4 Numerical benchmark of a non-linear elastic plate with hole 70
4.5 Conclusion 78

List of Figures **81**

Bibliography **85**

1 Introduction

Engineering is fundamentally the application of scientific principles for designing and creating systems, structures, and components. From tall buildings that reach the sky to the machinery we rely on daily, engineering remains a leading force in humanity's significant achievements. To understand how objects behave under acting forces, more than simple tests and basic models were required as the structures and materials we built became intricate, highlighting the need for better tools to predict the behavior of structures in various situations. In this regard, simulations have become an essential tool. They allow us to explore complicated systems in various situations. Such simulations are more than just beneficial; they are necessary, especially when full-scale experiments are too costly or time-consuming. In this setting, numerical simulations have become essential. Most of the methodologies developed since the dawn of modern numerical analysis have been preoccupied with the discretization of space and time. Finite differences, finite elements, finite volumes, molecular dynamics, and mesh-free methods are examples of different ways of estimating solution fields.

1.1 Computational solid mechanics

Solid mechanics is the branch of continuum mechanics that focuses on studying solid materials and their behaviors, particularly their motion and deformation when subjected to different forces, temperature changes, phase changes, and other external or internal factors. We can understand and accurately predict how materials and structures react to various forces through solid mechanics by developing mathematical models based on material behavior's physical principles. These models explain how materials experience deformation and how internal forces, known as stresses, are distributed throughout a material under strain. By applying concepts from areas such as elasticity, plasticity, and fracture mechanics, solid mechanics offers a comprehensive approach to simulate complex materials behaviors and ensure engineering designs' structural nature and functionality.

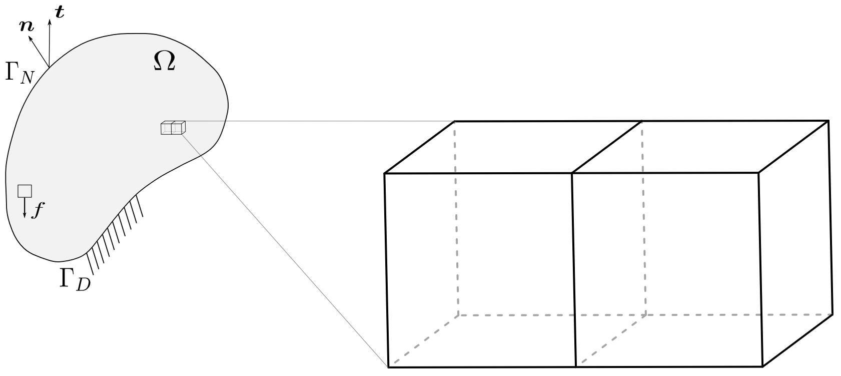
To begin with, we consider an elastic body $\Omega \subset \mathbb{R}^d$ with boundaries Γ_D and Γ_N such that $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$. In this context, a *material* refers to the substance or matter that constitutes the body Ω, i.e., various solids with varying properties and behaviors under different physical conditions. Central to the understanding of solid mechanics is the concept of *displacement*, represented by the field $\boldsymbol{u} : \Omega \to \mathbb{R}^d$, describing the movement of material points in a body from their original position to a new position. For a point $\mathbf{x} \in \Omega$, the displacement provides insight into how much and in which direction the point has moved.

Derived from the displacement field is the concept of *strain*, represented by the field $\boldsymbol{\varepsilon} : \Omega \to \mathbb{R}^{d \times d}_{\text{sym}}$, which quantifies the local deformation of the material and measures how it is stretched or compressed from its original configuration. The strain and its associated

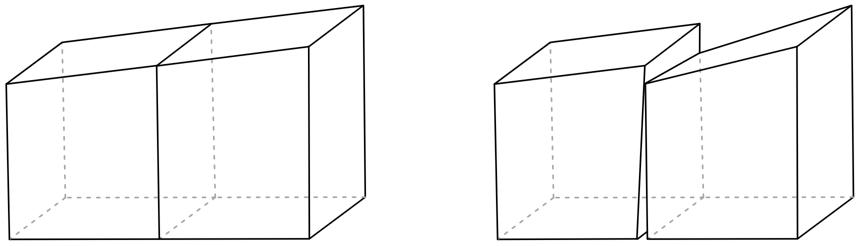
Dirichlet boundary condition are defined locally at any material point as

$$\begin{aligned} \boldsymbol{\varepsilon}(\mathbf{x}) &= \frac{1}{2}\left(\nabla \boldsymbol{u}(\mathbf{x}) + \nabla \boldsymbol{u}^T(\mathbf{x})\right), && \text{in } \Omega, \\ \boldsymbol{u}(\mathbf{x}) &= \boldsymbol{g}(\mathbf{x}), && \text{on } \Gamma_D. \end{aligned} \tag{1.1}$$

Here, the strain is defined by the symmetric gradient of the displacement field, called *compatibility*, constrained by essential boundary conditions with given displacement $\boldsymbol{g} : \Gamma_D \to \mathbb{R}^d$. The compatibility ensures the deformation is uniform and preserves continuity locally, cf. Fig. 1.1.1. Corresponding to strain is the concept of *stress*, represented by the field $\boldsymbol{\sigma} : \Omega \to \mathbb{R}^{d\times d}_{\text{sym}}$, which provides a measure of internal forces developed within a material as a reaction to applied external forces or deformations. The conservation principle in solid mechanics can be derived by considering an infinitesimal volume element, which ensures a balance of forces within the material. Thus, the *equilibrium* equation is encapsulated by the stress divergence and can be defined with the associated Neumann-type boundary condition by



(a) Undeformed microelement



(b) Compatible deformation

(c) Incompatible deformation

Figure 1.1.1: Deformation states of an infinitesimal volume element concerning strain compatibility. (a) Undeformed state where the volume element retains its original geometry. (b) Compatible deformation: showing uniform deformation and preserving continuity within the volume element. (c) Incompatible deformation: showing nonuniform deformation leading to discontinuities and potential internal stresses within the volume element.

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma}(\mathbf{x}) &= \boldsymbol{f}(\mathbf{x}), && \text{in } \Omega, \\ \boldsymbol{\sigma}(\mathbf{x}) \cdot \boldsymbol{n}(\mathbf{x}) &= \boldsymbol{t}(\mathbf{x}), && \text{on } \Gamma_N, \end{aligned} \tag{1.2}$$

where $\boldsymbol{f} : \Omega \to \mathbb{R}^d$ represents body forces acting on the material. The boundary condition on Γ_N accounts for external forces $\boldsymbol{t} : \Gamma_N \to \mathbb{R}^d$ along the outer normal direction $\boldsymbol{n} : \Gamma_N \to \mathbb{R}^d$ of the boundary. Fig. 1.1.2 demonstrates a three-dimensional infinitesimal element subjected to normal and shear stresses on each face. The stresses vary along the axes, as indicated by the derivatives regarding each axis. The illustration serves as a visual

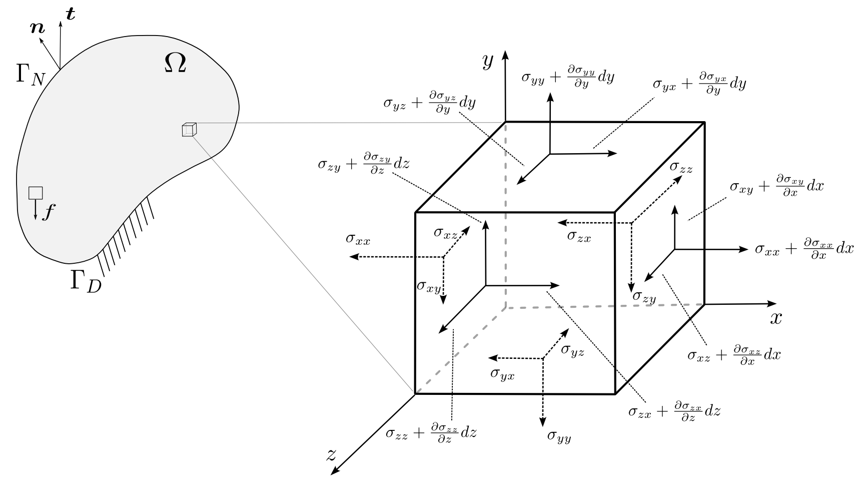

n
t
Γ_N
Ω
f
Γ_D
y
$\sigma_{yy} + \frac{\partial \sigma_{yy}}{\partial y} dy$
$\sigma_{yx} + \frac{\partial \sigma_{yx}}{\partial y} dy$
$\sigma_{yz} + \frac{\partial \sigma_{yz}}{\partial y} dy$
$\sigma_{zy} + \frac{\partial \sigma_{zy}}{\partial z} dz$
σ_{zz}
$\sigma_{xy} + \frac{\partial \sigma_{xy}}{\partial x} dx$
σ_{xx}
σ_{xz}
σ_{zx}
$\sigma_{xx} + \frac{\partial \sigma_{xx}}{\partial x} dx$
σ_{xy}
σ_{zy}
x
σ_{yz}
$\sigma_{xz} + \frac{\partial \sigma_{xz}}{\partial x} dx$
σ_{yx}
$\sigma_{zx} + \frac{\partial \sigma_{zx}}{\partial z} dz$
$\sigma_{zz} + \frac{\partial \sigma_{zz}}{\partial z} dz$
σ_{yy}
z

Figure 1.1.2: Illustration of the state of stress on an infinitesimal element within a continuum, demonstrating the conditions for static equilibrium by balancing the internal stresses.

aid in understanding the derivation of the equilibrium equations for a continuum, which asserts that the sum of the forces and moments on the element must be zero for the body to be in a state of static equilibrium. This condition is essential for analyzing and predicting the behavior of materials under various loading conditions in engineering and physics.

The combined state of strain and stress at any material point is described by $\boldsymbol{z} = (\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$, mapping the domain Ω to the local phase space $Z_{loc} \subset \mathbb{R}^{d\times d}_{\text{sym}} \times \mathbb{R}^{d\times d}_{\text{sym}}$, which provides a comprehensive view of the material's mechanical state. The collection of state functions $\boldsymbol{z} : \Omega \to Z_{loc}$ defines the global state space

$$Z := \{\boldsymbol{z} \,:\, \boldsymbol{z} \in Z_{loc}\}. \tag{1.3}$$

The compatibility and conservation laws are axiomatic and, therefore, material independent. Based on this we define a material-independent constraint set $\mathcal{C} \subset Z$ defined by

$$\mathcal{C} := \Big\{ \boldsymbol{z} \in Z : \text{(1.1) and (1.2)} \Big\}. \tag{1.4}$$

to be the set of states $\boldsymbol{z} = (\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ consistent with the compatibility and conservation laws and corresponding essential and natural boundary conditions thereof.

The constitutive equation encapsulates a material's intrinsic relationship between stress and strain. Since the material model is often unknown analytically, it is commonly approximated using a material data set $\mathcal{D} \subset Z$ defined by

$$\mathcal{D} = \{\boldsymbol{z} \in Z : \boldsymbol{z}(\mathbf{x}) \in \mathcal{D}_{loc}\}, \tag{1.5}$$

where $\mathcal{D}_{loc} = \{(\boldsymbol{\varepsilon}_i, \boldsymbol{\sigma}_i)\}_{i=1}^{n_e} \subset Z_{loc}$ is the local material data set derived from empirical strain-stress measurements in small-scale experiments. Here, $n_e \in \mathbb{N}$ is the number of local data points associated with the material point $\mathbf{x} \in \Omega$. To this end, the material data $\mathcal{D}$ is often completely replaced by the surrogate model $\mathcal{E}$ formally represented as

$$\mathcal{E} := \{\boldsymbol{z} \in Z \,:\, \boldsymbol{\sigma}(\mathbf{x}) = \mathcal{F}(\boldsymbol{\varepsilon}(\mathbf{x}))\}, \tag{1.6}$$

where $\mathcal{F} : \mathbb{R}^{d\times d}_{\text{sym}} \to \mathbb{R}^{d\times d}_{\text{sym}}$ is an approximated mapping, which maps strain to stress, characterizing the material's inherent properties and how they respond to deformation. For illustration purposes, we refer to Fig. 1.1.3. Given this relationship, we identify the material

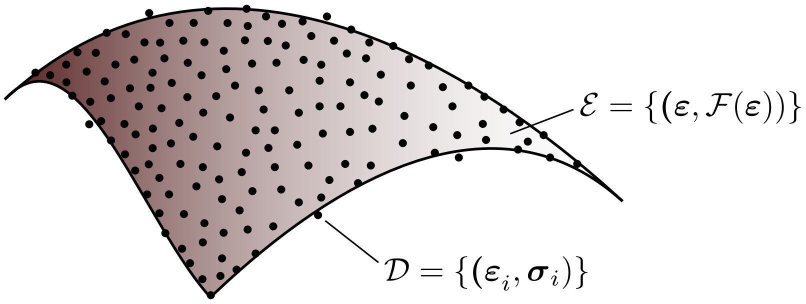

$\mathcal{E} = \{(\varepsilon, \mathcal{F}(\varepsilon))\}$
$\mathcal{D} = \{(\varepsilon_i, \sigma_i)\}$

Figure 1.1.3: Representation of the strain-stress relationship. The material model $\mathcal{E}$ (red, $-$) outlines the theoretical relationship based on the constitutive equation. The material data set $\mathcal{D}$ (black, $\cdot$) illustrates strain-stress measurements. The visualization emphasizes how the theoretical model is informed by, but distinct from, the experimental data

set $\mathcal{E} \subset Z$ as the collection of all possible states of the material that adhere to this specific constitutive relationship, irrespective of other constraints. The solution set of material states $z \in Z$ which comply with both compatibility (1.1) and equilibrium (1.1.2), and align with the material model (1.6) can be expressed as $\mathcal{C} \cap \mathcal{E}$.

1.1.1 Perspective on material modeling

Material modeling describes a solid material's relationship between strain and stress. Traditionally, material modeling has been approached from a macroscopic perspective, treating the material as a continuous medium and neglecting the underlying microstructure. These properties are determined through phenomenological relations, often derived from experimental data obtained through uniaxial or multiaxial tensile or compressive tests. These relations describe the material's bulk response, expressed through constitutive equations. For instance, a phenomenological model $\mathcal{E}$ can capture the relationship between the force applied and the resulting elongation when stretching a rubber band, mapping the macroscopic quantities of stress and strain tensors.

While phenomenological models provide a strong foundation, they have limitations, especially accounting for the influence of a material's microstructure on its properties. Microstructure refers to the arrangement of a material's fundamental building blocks, such as atoms, grains, and pores. For example, the size and orientation of grains within a metal can significantly impact its strength and ductility of a material. Recent advancements in computing power have enabled researchers to analyze a material's underlying microstructure, considering its atomic arrangements, grain boundaries, and deformations. By simulating the behavior of these microscopic components, researchers can predict the material's macroscopic response using techniques like micromechanics and crystal plasticity. Microscopic modeling offers several advantages. It allows for incorporating microstructure into the model, leading to a more accurate representation, especially for complex materials. Additionally, it provides insights into deformation and failure at the microscopic level, aiding the development of new materials with tailored properties.

The scope of material modeling extends beyond the foundational principles of elasticity and plasticity. For instance, it deals with complex areas such as damage and fracture mechanics, coupled time-dependent behaviors of viscoelasticity and viscoplasticity, and the influence of environmental factors like temperature and humidity on materials. Accurately capturing such complex phenomena demands a profound understanding of the material's macro and microscopic properties. Unfortunately, this task becomes more challenging, especially with the progression of experimental techniques. Therefore, the fitting procedure requires more advanced calibration of material parameters, leading to a combination of extensive experimental data with advanced computational techniques. Consequently, the field is moving toward data-driven methods such as machine learning, Bayesian methods, and materials informatics. These methods allow for precise predictions and offer improvements over traditional modeling techniques. The ongoing goal in material modeling is to increase the accuracy and range of these models. As technology advances, material modeling is expanding its potential, not just in predicting behavior but also in developing new materials.

1.1.2 Integration of machine learning in model-based approaches

Machine learning is revolutionizing computational mechanics by empowering researchers to interpret complex data patterns beyond the capabilities of traditional modeling. The new techniques lead to more advanced material modeling, improved predictions of material behavior, and streamlined manufacturing processes.

Notably, through Supervised Learning [1], models are trained on datasets with known outcomes. For instance, polynomial regression, a type of Supervised Learning, fits complex, non-linear equations to data, effectively capturing the intricate relationships within material responses. It constructs polynomial features of the input variables, adeptly modeling the interaction between strain and stress [2]. Support Vector Machines (SVMs, [3]) elevate this process by finding the optimal hyperplane that divides datasets into classes. SVMs work by constructing a high-dimensional feature space and then identifying the hyperplane that best separates the classes, a technique particularly useful for classifying materials based on various attributes such as phase or property [4].

Conversely, Unsupervised Learning algorithms like K-means [5] and hierarchical clustering [6] organize data based on inherent similarities without needing predefined labels. K-means clusters data by minimizing variance within each cluster, which could categorize materials based on microstructural features [7, 8]. Hierarchical clustering, in contrast, builds a tree of clusters, offering a multilevel hierarchy that is beneficial for dissecting complex material properties [9]. Principal Component Analysis (PCA, [10]) simplifies the data analysis process by reducing the number of variables. It transforms a large set of variables into a smaller one that still contains most of the original dataset's information. It is particularly advantageous for analyzing high-dimensional material science data [11].

Neural Networks such as Feed-Forward (FFNN, [12]), Convolutional (CNN, [13]), and Recurrent Neural Networks (RNN, [14]) leverage different structures to model complex functions. FFNNs use layers of nodes to approximate non-linear relationships [15], CNNs apply convolutional layers to parse spatial information in data like material microstructure images [16], and RNNs use loops within their architecture to process sequential data, such as time-series information on material fatigue [17].

Reinforcement Learning algorithms [18] learn through interaction with an environment, seeking to maximize a reward function, which can be harnessed to find optimal material processing conditions [19]. Deep Learning techniques like Autoencoders [20] and Generative Adversarial Networks (GANs, [21]) also contribute by encoding data into compact representations for feature extraction [22] and generating new, synthetic material data for extensive analysis [23]. In addition, Autoencoders offer a unique ability to learn compressed representations of material data, which can be helpful for dimensionality reduction and anomaly detection in material properties. Graph Neural Networks (GNNs, [24]) treat the material as a graph, capturing the interconnectivity between elements, which is crucial for predicting new material properties [25]. Transfer Learning [26] and Bayesian Optimization [27] also play a part, with the former reapplying learned models to new tasks [28] and the latter strategically exploring and optimizing material designs using probabilistic models.

To summarize, machine learning techniques collectively enhance material science, allowing for the intricate modeling of material behavior, advanced classification of material properties, and innovative approaches to data analysis and generation. They introduce significant advancements in materials research and development by capturing complex relationships within materials, optimizing processing parameters, and generating novel materials through synthetic data.

1.2 Model-free data-driven computational mechanics

In computational mechanics, not every solution is straightforward. Even with a clear understanding of material behaviors by a model $\mathcal{E}$, complexities can arise with the satisfaction of the differential equation-based requirement set $\mathcal{C}$. The intersection between these sets may be empty, implying that possibly no point within the material model (1.6) aligns with the compatibility (1.1) and equilibrium (1.2) principle; i.e. $\mathcal{C} \cap \mathcal{E} = \emptyset$.

A possible resolution is to introduce a *relaxation*, cf. [29], of the definition of the solution set $\mathcal{C} \cap \mathcal{E}$. Instead of seeking a direct intersection, we consider the closest distance between these sets within a predefined metric space, denoted as $d : Z \times Z \to \mathbb{R}$. The solution, thus, can be mathematically represented as:

$$\underset{\boldsymbol{z} \in \mathcal{C}}{\arg\min}\, d(\boldsymbol{z}, \mathcal{E}). \tag{1.7}$$

In addressing the minimization problem described by (1.7), we focus on a specific mapping designed to negotiate material model points with the constraint set $\mathcal{C}$. For a given material state $\hat{z} = (\hat{\varepsilon}, \mathcal{F}(\hat{\varepsilon})) \in \mathcal{E}$ the mapping $P_{\mathcal{C}} : \mathcal{E} \to \mathcal{C}$ defined as:

$$\underset{\boldsymbol{z} \in \mathcal{C}}{\arg\min} \left\{ d((\boldsymbol{\varepsilon}, \boldsymbol{\sigma}), (\hat{\varepsilon}, \mathcal{F}(\hat{\varepsilon}))) - [(\nabla \cdot \boldsymbol{\sigma} - \boldsymbol{f}) \cdot \boldsymbol{\eta} + (\boldsymbol{\varepsilon} - \nabla^{\text{sym}} \boldsymbol{u}) : \varsigma] \right\}, \tag{1.8}$$

identifies the nearest point $\boldsymbol{z}$ within the constraint set, cf. Fig 1.2.1a. In regard, $\boldsymbol{\eta}$ and ς, representing Lagrange multipliers, are introduced to ensure that the solution satisfies the physical constraints.

Corresponding to $P_{\mathcal{C}}$, the complementary mapping $P_{\mathcal{E}} : \mathcal{C} \to \mathcal{E}$ given by

$$\underset{\hat{\boldsymbol{z}} \in \mathcal{E}}{\arg\min}\, d((\boldsymbol{\varepsilon}, \boldsymbol{\sigma}), (\hat{\varepsilon}, \mathcal{F}(\hat{\varepsilon}))), \tag{1.9}$$

determines the closest point $\hat{\boldsymbol{z}}$ in the material set $\mathcal{E}$ for a given state $\boldsymbol{z} = (\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ within the constraint set $\mathcal{C}$, cf. Fig. 1.2.1b.

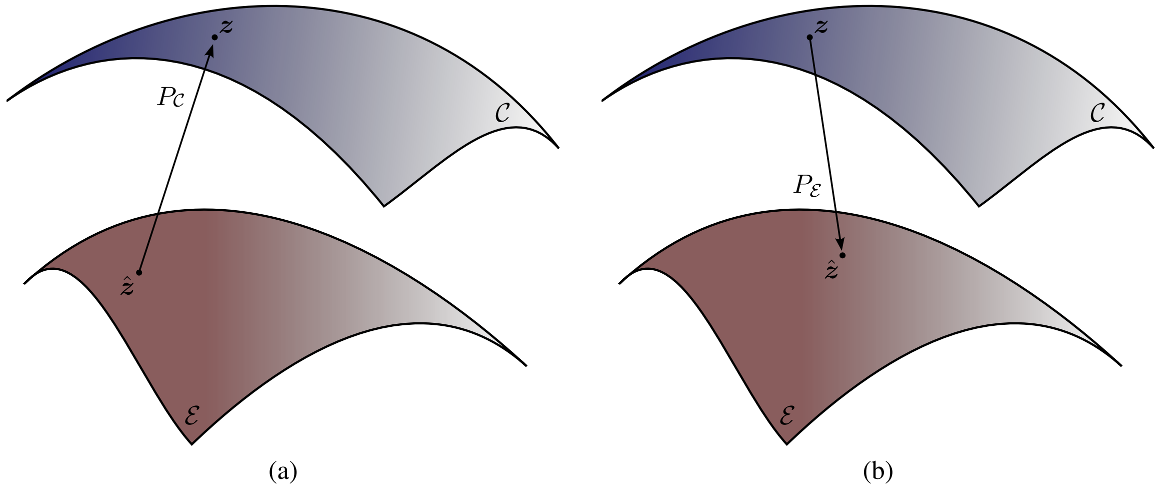


Figure 1.2.1: Visualization of the mappings $P_{\mathcal{C}}$ and $P_{\mathcal{E}}$: (a) demonstrating the projection of a point $\hat{\boldsymbol{z}}$ in the material model $\mathcal{E}$ onto the nearest point $\boldsymbol{z}$ within the constraint set $\mathcal{C}$. (b) showcasing the reverse mapping from the constraint set to the material set, identifying the nearest material state $\hat{\boldsymbol{z}}$ to a given constraint state $\boldsymbol{z}$.

To effectively find the nearest points between the constraint set and the material model and, therefore, solve the minimization problem (1.7), a fixed-point technique is utilized:

$$\hat{\boldsymbol{z}}^{k+1} = P_{\mathcal{E}}(P_{\mathcal{C}}(\hat{\boldsymbol{z}}^k)), \tag{1.10}$$

where $k \in \mathbb{N}$ represents the iteration step. The iteration stops if the states converge, i.e., the same states are calculated, or a predefined distance threshold is reached. An illustration of the fix-point approach is shown in Fig. 1.2.2.

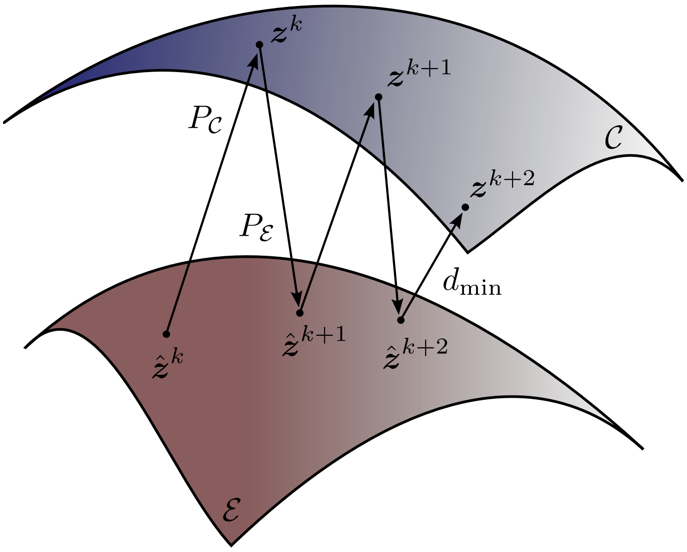


Figure 1.2.2: Visualization of the fixed-point approach for the relaxed solution scheme (1.7). Iteration between projections: $P_{\mathcal{C}}$, which maps data points $\hat{\boldsymbol{z}} \in \mathcal{E}$ to the closest constrained points $\boldsymbol{z} \in \mathcal{C}$, and the projection $P_{\mathcal{E}}$, which locates closest point in the material set to the constrained state provided by $P_{\mathcal{C}}$. The iteration is visualized by $\{(\hat{\boldsymbol{z}}^k, \boldsymbol{z}^k), (\hat{\boldsymbol{z}}^{k+1}, \boldsymbol{z}^{k+1}), \ldots\}$, converging to minimum distance d_{min}.

A notable instance of such a relaxed formulation is the *model-free data-driven computational mechanics* approach introduced by Kirchdoerfer and Ortiz [30]. Their methodology aims to determine the state z in the constraint set $\mathcal{C}$ closest to the material data set $\mathcal{D}$, establishing a connection that best captures the constitutive representation of an unknown material model $\mathcal{E}$. The primary benefit of this method is that it bypasses the traditional material modeling step altogether, paving the way for potentially more direct and accurate solutions for even very complex materials. A straightforward representation of model-free data-driven computing determines the measure of deviation as the distance between $\boldsymbol{z}$ and $\mathcal{D}$. From this, we can outline the corresponding data-driven distance-minimization problem as

$$\arg\min_{\boldsymbol{z}\in\mathcal{C}} d(\boldsymbol{z}, \mathcal{D}). \tag{1.11}$$

To find the closest point in the constraint set to the material data set, Kirchdoerfer and Ortiz use the fixed-point approach (1.10), denoting it by

$$\hat{\boldsymbol{z}}^{k+1} = P_{\mathcal{D}}(P_{\mathcal{C}}(\hat{\boldsymbol{z}}^{k})), \tag{1.12}$$

where in this case $\hat{\boldsymbol{z}} \in \mathcal{D}$ describes a point in the data set. The first mapping $P_{\mathcal{C}} : \mathcal{D} \to \mathcal{C}$ projects a data state $\hat{\boldsymbol{z}} = (\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}})$ to the closest point $\boldsymbol{z} = (\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ in the constraint set. Similar to Eq. (1.8), the projection at iteration k can be stated for fixed data points $\hat{\boldsymbol{z}}$ as

$$\arg\min_{\boldsymbol{z}\in\mathcal{C}} \left\{ d((\boldsymbol{\varepsilon}, \boldsymbol{\sigma}), (\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}})) - \left[(\nabla \cdot \boldsymbol{\sigma} - \boldsymbol{f}) \cdot \boldsymbol{\eta} + (\boldsymbol{\varepsilon} - \nabla^{\text{sym}} \boldsymbol{u}) : \boldsymbol{\varsigma} \right] \right\}. \tag{1.13}$$

One should notice that in the original framework [30], the compatibility condition (1.1) is directly substituted into the distance function, and only the equilibrium condition (1.2) is enforced through the multiplier approach. In a finite-element framework, equation (1.13) results in two linear equation systems that must be solved for actual and virtual displacements $\boldsymbol{u}$ and $\boldsymbol{\eta}$. Governing equations then recover the corresponding material strain-stress state $\boldsymbol{z} = (\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$. The second projection $P_{\mathcal{D}} : \mathcal{C} \to \mathcal{D}$ finds the closest state in the data set to the previously calculated state in the constraint set by

$$\arg\min_{\hat{\boldsymbol{z}}\in\mathcal{D}} d((\boldsymbol{\varepsilon}, \boldsymbol{\sigma}), (\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}})). \tag{1.14}$$

Generally, this is done by calculating the nearest neighbors in the local data sets for each material point. Notably, the data set does not differentiate between stress and strain, treating them, and other history variables equally. Similar to solving an unsupervised regression, we attempt to match the current state in the constraint set to a state in the data set. As mentioned before, solutions obtained by this data-driven paradigm depend only on the material data set $\mathcal{D}$. Thus, no effort is directed towards material modeling or approximating the local data sets. The following section will overview recent studies and advancements in this data-driven methodology. Fig. 1.2.3 shows the data-driven paradigm using only the data points from small-scale measurements to find the closest material state in the constraint set.

1.2.1 State-of-the-art

Elasticity:
Elasticity modeling has seen significant changes with increasing computational capabilities and data availability. Kirchdoerfer and Ortiz [30] introduced a data-driven approach to

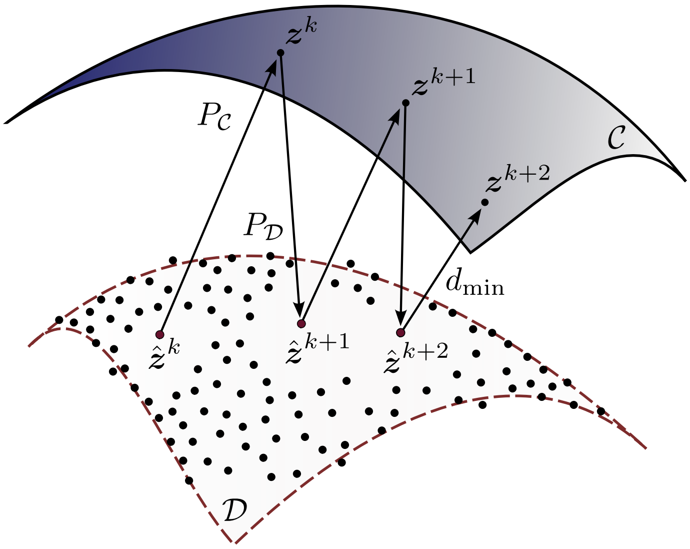


Figure 1.2.3: Visualization of the fixed-point approach for the data-driven relaxed solution scheme (1.11), adapted to minimize the distance to dataset $\mathcal{D}$. Iteration between projections: $P_{\mathcal{C}}$, which maps data points $\hat{z} \in \mathcal{D}$ to the closest constrained points $z \in \mathcal{C}$, and the projection $P_{\mathcal{D}}$, which locates the closest point in dataset $\mathcal{D}$ to the state provided by $P_{\mathcal{C}}$. The iterative process is depicted by the sequence of points $\{(\hat{z}^k, z^k), (\hat{z}^{k+1}, z^{k+1}), \ldots\}$, converging to minimize the distance d_{min}.

elasticity problems. This method incorporates experimental material data into conservation laws, eliminating the need for traditional empirical material modeling. Applications of this method in non-linear truss mechanics and linear elasticity showed results in terms of convergence and robustness. Conti et al. [29] further investigated the concept, provided a deep mathematical formulation, and demonstrated its convergence in elasticity. In addition, they noted challenges in handling noise but suggested that more extensive and complete datasets improve the solutions. The work also emphasized the capacity of data-driven models to interpret material data in ways not limited to traditional graphs. Expanding the methodology, Kanno [31] combined the robustness of mixed-integer programming with data-driven elasticity. This method was designed to be compatible with standard mixed-integer solvers and showed potential as a reference for future algorithms. Transitioning to more specific problem areas, RÃ¶ger and Schweizer [32] focused on scalar elliptic problems in a data-driven context. This work analyzed the relaxation dynamics of datasets with one outliner. Kirchdoerfer and Ortiz [33] resolved the issue of noise and outliers by proposing the max-ent data-driven paradigm. By exploiting clustering analysis, the model effectively weighted data points based on their relevance to the final solution, enhancing the method's stability. The original minimum-distance solver is restored as the cluster size decreases to one point. Sanz et al. [34] emphasized the utility and relevance of the data-driven approach in tissue biomechanics. Due to their unique microstructural and mechanical variations, biological tissues often challenge traditional constitutive models. By applying the data-driven method to cortical bone tissue, they captured the complex characteristics of bone tissue, such as its heterogeneity and intricate architecture. The study asserted the potential of the data-driven method in simulating biomechanical scenarios without the need for complex spatial models.

Finite strain:
The work of Conti et al. [29] explores finite elasticity within the data-driven framework, particularly extending from their prior findings in geometrically linear elasticity. Their in-

vestigation centered on establishing a robust framework where the issues of finite elasticity are well-posed in terms of solution existence. Setting their study within deformation gradients and the first Piola-Kirchhoff stresses, they introduced and expanded upon notions of coercivity and material data set closedness. They were able to prove the existence of solutions and illustrate this with examples adhering to material frame indifference. With a deeper understanding of these mechanisms, Platzer et al. [35] evaluated their practical application in finite strain elasticity. By employing a Lagrangian finite element formulation combined with a novel resolution scheme, they predicted global responses of structures using sparse and optimal databases. The findings revealed that even if compact, optimal datasets had a distinct advantage in predicting local strain and stress states, reinforcing that problem-specific tailored material databases enhance data-driven simulations. In [36], Platzer studied the complex relationship between data consumption and production in computational mechanics. He extended the data-driven approach to a geometrically non-linear setting and proposed a method for generating databases through mechanically meaningful sampling. Building on this foundation, Platzer et al. [37] presented a finite element solver for data-driven finite strain elasticity. Noteworthy is their introduction of an efficient tree-based nearest neighbor search algorithm and an emphasis on maintaining angular momentum balance for the method's convergence. In addition, Nguyen and Keip [38] discussed non-linear elasticity. The research advanced the concept of the distance-minimizing data-driven computing method by making it applicable to boundary-value problems in continuum mechanics under finite strain theory. They established a foundation for finite element formulations, highlighting the structural tangent stiffness and generalized force vectors. Thus, Nguyen et al. [39] presented a structured variational framework for the distance-minimizing data-driven method. By segmenting the double-minimization problem into two distinct single-minimization problems, they formulated one as a constrained optimization problem with solutions sought using Lagrange multipliers. Emphasis was laid on the potential of spectral element methods, underscoring their capability to reduce computational cost without compromising solution accuracy.

Damage and fracture:
Developing reliable and efficient computational methods for fracture mechanics remains crucial for design choices, improving safety standards, and optimizing material utilization. Carrara et al. [40] made significant advancements in brittle fracture mechanics by presenting a data-driven approach distinct from conventional material modeling techniques. Their method combines governing equations from variational principles with discrete data points. The idea is to identify a solution within the data set that either aligns with the Kuhn-Tucker conditions derived from the variational fracture problem or minimizes a specific energy function. This dual-method approach was tested under various configurations, proving its robustness for Griffith and R-curve type fracture behaviors, even with noise. Building upon their initial framework, Carrara et al. [41] extended the data-driven paradigm to rate-dependent fractures and sub-critical fatigue. The authors integrated balance governing equations, embedded in variational principles, with a set of data points characterizing the fracture behavior of materials. The quality of their solution lies in identifying the data point that aligns best with the stability condition. Various tests confirmed the method's applicability across multiple rate-dependent fracture and fatigue models, irrespective of noise factors.

Dynamics:
Kirchdoerfer and Ortiz [42] extended the data-driven paradigm with a specific focus on incorporating time integration into both distance-minimizing and entropy-maximizing schemes.

This study introduces a paradigm where data points are assigned relevance based on their proximity to the solution and their maximum-entropy weighting. The result leads to schemes that minimize specific free energy, considering the phase space and a time-discretized momentum conservation constraint. Transitioning from dynamics, Gonzalez et al. [43] studied the thermodynamic consistency within data-driven computational mechanics. The study offers an approach that complies with conservative and dissipative dynamics components while ensuring compliance with fundamental principles like energy conservation and positive entropy production.

Manifold learning & reduction techniques:
Ibañez et al. [44] showcased the potential of constructing a complete constitutive manifold by focusing on several complex experimental tests. This approach offers a strategic method for managing extended data requirements, providing a more efficient simulation process without relying on traditionally complex models. In [45], they build upon this by introducing a novel data-driven method embedded in manifold learning methodologies. The research illustrates a way to extract essential information from comprehensive datasets. Regarding manifold learning, the study by Gonzalez et al. [46] investigated the challenge of hidden model parameters, especially in contexts with complex microstructures. The paper formulates a solution adept at handling high-dimensional models by utilizing the kernel Principal Component Analysis to discern these elusive parameters and coupling them with the Proper Generalized Decomposition. Combining these techniques offers a method to deal with high-dimensional spaces effectively. In addition, Bahmani and Sun [47] presented a new approach with the manifold embedding technique tailored for elasticity problems. They use an invertible neural network to embed constitutive data into an Euclidean space. This approach simplifies the local distance-minimization problem and improves the consistency of the numerical solution, especially with limited or incomplete data sets. Another approach, introduced by Eggersmann et al. [48], expands the data-driven computing method by integrating locally linear tangent spaces into datasets. By employing the tensor voting method, the research enhances the precision and understanding of data structures. This method not only deals with noisy datasets but also optimizes the accuracy of the computational results.

Inelasticity
Eggersmann et al. [49] expanded the data-driven approach to inelasticity, which was crucial, given that material behavior can evolve in time and, thus, be history-dependent. They explored several paradigms to represent the evolving material datasets, ranging from materials with memory over differential materials to history variables. Considering the combination of the paradigms, this research offers flexibility to choose an approach best aligned with the specific inelastic material type. The study also showcased the potential of data-driven inelasticity through numerical examples in capturing complex behaviors with improved computational efficiency. Bahmani and Sun [50] offered a combination of traditional and data-driven methodologies to deal with poroelastic materials. They introduced three formulations to address the interactions within fluid-infiltrating porous media, each varying based on data availability. A feature was an efficient distance-minimized algorithm, which utilized a k-dimensional tree search to accelerate the process. The model also acknowledged the variances between solid elasticity and fluid hydraulic data. In addition, Salahshoor and Ortiz [51] presented a framework focused on viscoelastic wave propagation. The method is formulated in the frequency domain. It capitalizes on the advantages of the flat-norm of the Fourier transform, allowing the system's response to be inferred across varying frequencies. The robustness and compatibility of this scheme were underscored through its

application to real-world materials, such as polymeric truss structures and soft gel samples. In contrast, Bartel et al. [52] addressed the difficulties of path-dependent material behavior by introducing the concept of history surrogates and propagators. A notable benefit of this method is its integration with pre-existing algorithms, enabling a smooth transition and the flexibility to address heterogeneous problems that comprise both elastic and elastoplastic materials. Lastly, Poelstra et al. [53] focused on plasticity, proposing an evolutionary data-driven scheme. The study provided a comprehensive analysis of existence and data convergence.

Coupled problems:
Yang et al. [54] proposed an approach that uses the data-driven paradigm in local regions where determining material constitutive models is challenging. Simultaneously, the model-driven method is employed in regions where material models are available. Marenic et al. [55] addressed the challenges of multiphysics. Traditional methodologies to solve such problems necessitate creating monophysical and multiphysics coupling operators. Marenic's work simplified this by introducing a data-driven approach tailored for 'smart' materials. The idea is to expand the phase space and formulate a new norm specific to the data-driven solver. The study illustrated that bypassing the coupling tangent terms is feasible by effectively leveraging a comprehensive material database that inherently captures coupling interactions. The data-driven method allows for decoupling different physical aspects while managing their interactions through data.

Multiscale framework:
Karapiperis et al. [56] addresses the issue of nonlocal effects, which inherently exist in microstructured materials such as metals, foams, and granular media. Although there have been continuous efforts to solve predictive nonlocal mechanical theories, a definitive internal length scale has yet to be achieved. Karapiperis introduces a framework that eliminates the necessity of defining any internal length scale using the data-driven paradigm. This paradigm is expanded to generalized continua. Two applications, including a micropolar elastic plate and the shear banding problem in quartz sand, underline the method's robustness and effectiveness. Xu et al. [57] optimized element methods, particularly for composite materials and structures. The paper introduces the data-driven FE^2 method, highlighting the computations of correlated scales observed in classical FE^2. By computing microscopic problems in advance, they create an offline material genome database. This database subsequently plays a role in the macroscopic data-driven analysis, overcoming issues associated with multiscale systems. The approach replaces the real-time computation of microscopic problems with a more efficient database-driven approach, significantly increasing the computational efficiency of structural analysis. Karapiperis et al. [58] emphasized material history parametrization and optimal sampling of the mechanical state space. A focus on the behavior prediction of sand, which is inherently complex and history-dependent, underscores the model's effectiveness. The model offers predictions under intricate nonmonotonic loading paths and showcases parallels with plane strain and triaxial compression shear banding experiments. Mora et al. [59] implemented a multiscale data-driven approach to study cortical bone tissue, showing that the data-driven paradigm suits real-world problems. They obtained detailed strain fields at microscopic and macroscopic levels through biaxial loading of horse cortical bone samples and subsequent digital image correlation. Their findings at the macroscopic level highlighted nonuniform strain and stress patterns, questioning the suitability of using a standard linear homogeneous orthotropic model for bone tissue.

Uncertainty:

Guo et al. [60] explicitly incorporated an uncertainty analysis-based framework into data-driven computational mechanics. Instead of restricting the analysis to a single solution trajectory in phase space, it seeks a comprehensive solution set. The aim is to discern the effects of multisource uncertainties linked with the data set on the resultant data-driven solutions. In addition, Zschocke et al. [61] highlighted that materials like concrete or reinforced concrete inherently exhibit behavior dictated by their heterogeneities. The study translates heterogeneous mesoscale behavior into the macroscopic continuum by tapping into numerical homogenization methods, which embed themselves in scale separation. The result is a decoupled numerical homogenization scheme encapsulating material uncertainties within a singular dataset. Korzeniowski [62] introduced computations driven by data generated from stochastic fields. A highlight is the introduction of a multilevel computation technique designed to optimize numerical efforts. Instead of using complete data sets, the approach employs multiple simulations guided by adaptive, compact data subsets. Korzeniowski extended the data-driven methodology's applicability, examining its capacity by addressing a diffusion problem and integrating a fuzzy variable to build a polymorphic uncertainty model.

Material (parameter) identification:
Leygue et al. [63] research identified the strain-stress relationship of non-linear elastic materials. The data-driven approach builds a robust database by leveraging collections of non-homogeneous strain fields, such as those measured from Digital Image Correlation (DIC). Truss structures serve as the starting point for this methodology, eventually widening its scope to enclose small-strain elasticity. In another study, Leygue et al. [64] introduced the inverse data-driven approach based on full-field measurements. This technique permits the extraction of heterogeneous and multiaxial material state fields directly from displacement fields and external load measurements, bypassing the constraints of traditional constitutive equations. Moreover, Stainier et al. [65] focused on identifying material data through measures of displacement fields. The core Data-Driven Identification (DDI) algorithm determines compatible mechanical states, the material database, and the underlying structural response, enabling accurate predictions. Dalemat et al. [66] captured heterogeneous stress fields by coupling the DIC and the DDI algorithm. The approach retrieves the material's mechanical response from a rich database of displacement fields and loading conditions based solely on mechanical equilibrium. Based on this, Dalemat et al. [67] investigated the data-driven identification algorithm's resilience against incomplete input data. By ingeniously recovering missing data and only ensuring the balance equation, Dalemat et al. emphasized the robustness of this approach, even in the face of missing displacement values and partial force information. Su et al. [68] augmented the data-driven identification paradigm by integrating the locally convex reconstruction method, named the Local-Convexity Data-Driven Identification (LCDDI) approach. The method optimizes material data points considering local data structures, offering significant improvements in data acquisition and quality in material identification.

Stochastic:
Ayensa et al. [69] developed a method using the Mahalanobis distance to address engineering scenarios where data exhibit variance and correlation. When tested on various problems, their approach showed improved convergence, accuracy, and flexibility performance, especially for noisy data. Beyond this, Korzeniowski and Weinberg [70] focused on situations with sparse data. In these cases, maintaining the robustness of numerical simulations becomes crucial. The authors compared two methods â€“ the stochastic and the data-driven finite element methods. Through a comparative analysis, Korzeniowski showed the advantages of the data-driven finite element method, especially when dealing with uncertain

material data. Prume et al. [71] extended the data-driven methodology, proposing using likelihood measures based on empirical data sets to estimate the potential outcomes of a structure. This method avoids conventional modeling and incorporates efficient techniques like population annealing and efficient search algorithms. Conti et al. [72] introduced a method that derives inferences directly from empirical data. The paper introduces a novel intersection concept between likelihood measures, disclosing how to establish system outcomes' likelihood. It derives explicit analytic expressions for outcome expectations by combining entropic regularizations and empirical data sets.

Efficiency:
Eggersmann [73] explored the computational aspects of the data-driven paradigm, precisely the challenge of searching for the closest matches in an extensive material data set. Various data structures were assessed for their efficiency in this task. The findings were that using approximate nearest-neighbor algorithms resulted in computational improvements without affecting the accuracy of solutions. The concept by Van et al. [74] utilized unprocessed data, which consist of displacement fields across extensive areas. This data is collected from full-field measurements and comes from real-world structure measurements. They are reliable for understanding movement and stress in structures. Collecting and studying the data makes it possible to guess how new structures might react.

1.2.2 Integration of machine learning in model-free approaches

As explored in subsection 1.1.2, integrating machine learning with traditional model-based methods in computational mechanics indicates a promising approach, offering innovative resolutions to long-established challenges. Techniques like Neural Network Regression and Bayesian Optimization enhance prediction accuracy while learning paradigms such as Supervised, Unsupervised, and Reinforcement Learning, alongside Hybrid Models, advance the traditional methodologies. The pertinent question is: *How can machine learning be instrumental for model-free methods?*

Unlike model-based approaches, model-free methods bypass the preliminary step of material modeling, instead directly utilizing material datasets to solve mechanics problems. This direct application is particularly beneficial with high-quality data. However, it encounters noisy, sparse, or extensive data issues, which worsen when the material behavior exhibits non-linearity or plasticity. Machine learning's prowess in pattern recognition and predictive analytics is pivotal, offering the potential to manage such datasets adeptly.
For instance, Kalman filtering [75] serves as an iterative enhancement tool that refines predictions by effectively accounting for noise within datasets, all without reliance on physical models. This method excels by capitalizing on the statistical characteristics intrinsic to the data. Similarly, when confronted with the task of efficient data querying in extensive datasets, KD-trees [76] provide an invaluable means to partitioning nearest-neighbor searches within a multi-dimensional space, thus addressing the optimization challenge outlined by Eq. (1.14). Sparse data often presents a significant issue in model-free analysis; here, dimensionality reduction techniques such as Principal Component Analysis (PCA, [10]) and t-Distributed Stochastic Neighbor Embedding (t-SNE, [77]) can be transformative. These techniques refine the nature of the data, emphasizing informative trends and patterns crucial for the analysis, thus facilitating a clearer understanding of complex datasets.

Furthermore, Physics-Informed Neural Networks (PINNs, [78]) and Particle Swarm Optimization (PSO, [79]) exemplify the synergy between machine learning and computational mechanics optimization tasks. PINNs adeptly incorporate physical laws as constraints within the learning process, ensuring that the predictive model adheres to known principles. The intrinsic characteristics of Eq. (1.13) outline a loss function inherently compatible with PINNs. Using neural networks facilitates the inclusion of supplementary conditions directly within the loss function, embedding more complex constraints and enhancing the model's fidelity to physical laws. Concurrently, PSO harnesses a group's collective intelligence to navigate the solution space, proving particularly adept when the data is noisy or the physical laws are incompletely understood. It employs multiple agents or ”particles” that explore the search space and update their trajectories based on their own experience and that of their neighbors. Its effectiveness arises from its collaborative approach and random processes, which empower it to identify the best overall solutions even in environments with local optima.

In conclusion, machine learning applied to computational mechanics facilitates more accurate predictions and enhances computational efficiency. It also introduces adaptability to new and complex data, strengthening model-free analysis's robustness. Machine learning is a suitable tool for an increasingly data-driven engineering field that can redefine problem-solving paradigms.

1.3 Outline of the enclosed articles

Article 1 – Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules: The traditional model-free data-driven paradigm has been effective in addressing complex non-linear elastic material behavior. However, it faces significant challenges when dealing with materials with history-dependent properties. In these materials, a single point in the stress-strain space can correspond to different material behaviors, depending on the loading history. The article in chapter 2 introduces a novel mixed-method approach that integrates the traditional model-based framework with the emergent data-driven paradigm. The method augments the dataset with experimentally measured directions in the tangent space of points in stress-strain space (cf. Fig. 1.3.1), capturing potential changes during loading. This information is crucial for differentiating between elastic and inelastic behavior, as elastic materials exhibit a constant tangent direction throughout loading. In contrast, inelastic materials display changing tangents due to their non-linear response. These tangents enrich the data with crucial information about the material's history-dependent behavior.

Moreover, the dataset is divided into subsets, each correlating to distinct behavioral patterns of the material. A set of transition rules are defined to map between the subsets. The efficiency of this method is demonstrated through its application to non-linear elastic and elasto-plastic behavior with isotropic hardening, thereby emphasizing its potential for the analysis and prediction of material behavior under complex loading conditions.

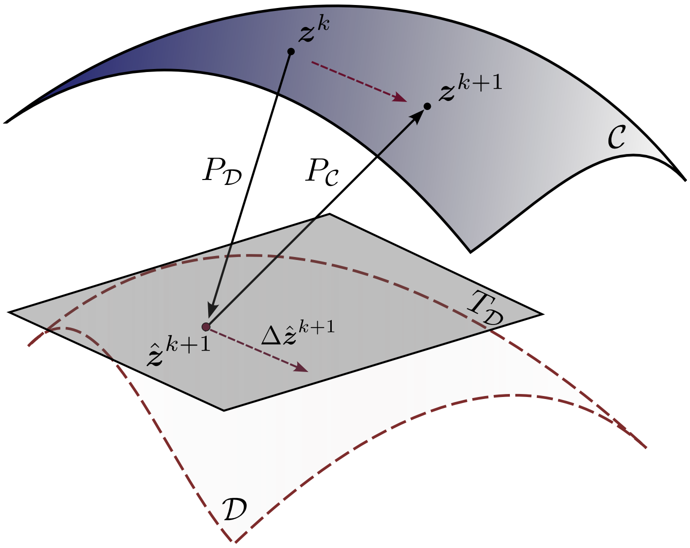


Figure 1.3.1: The distance minimization algorithm iterates between two steps: first, local projection $P_\mathcal{D}$ maps the material point $\boldsymbol{z}^k$ from the conservation manifold $\mathcal{C}$ onto the closest data point $\hat{\boldsymbol{z}}^{k+1}$ within the data set $\mathcal{D}$. Projection $P_\mathcal{C}$ maps the point back onto the resulting equilibrium state $\boldsymbol{z}^{k+1}$ in $\mathcal{C}$, incorporating the associated tangential space $T_\mathcal{D}$ to ensure that the change in material state between loading steps aligns with the data tangent $\Delta\boldsymbol{z}^{k+1}$.

Article 2 – Model-free data-driven inelasticity in Haigh-Westergaard space – a study how to obtain data points from measurements: From the standpoint of practical application, one of the critical aspects of the tangent space-enhanced model-free data-driven approach is the comprehensive coverage of the relevant data states, which involves collecting qualified data and computing the corresponding tangent spaces for each point. Leveraging material symmetry is beneficial in this context, as it effectively reduces the data requirements without sacrificing the quality of the simulation outcomes. The study in chapter 3 develops a methodology using Haigh-Westergaard coordinates, illustrated in Fig. 1.3.2, to facilitate an understanding of the material's yield surface. Regarding this, we utilize a dual experimental setup: a combined tension-torsion test to determine the yield surface and a single tensile test to derive the tangent space at specific points. This approach provides a comprehensive data set that accurately reflects the material's response under different loading conditions. The proposed data-driven approach aims to minimize the distance within the Haigh-Westergaard space, incorporating tangent space directions while adhering to compatibility and equilibrium constraints. This refined paradigm accurately represents the behavior of materials under elasto-plastic deformation with isotropic hardening.

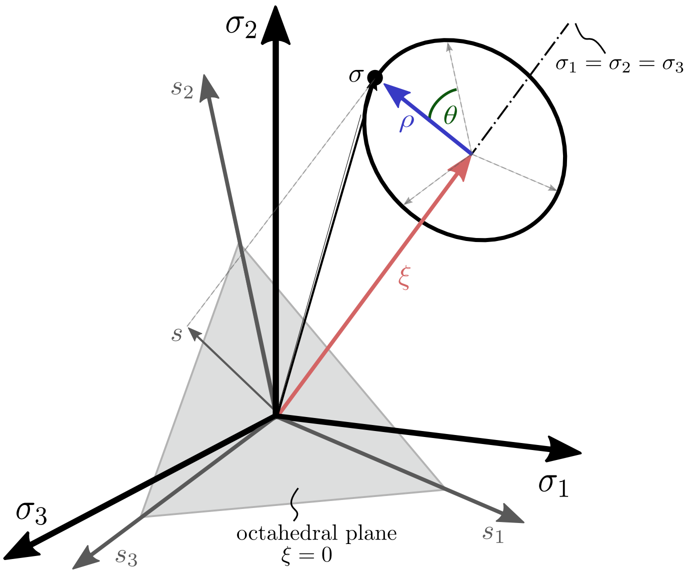


Figure 1.3.2: Visualisation of stress $\boldsymbol{\sigma}$ and its deviatoric part $\boldsymbol{s}$ in the Haigh-Westergaard stress space. The coordinates are defined by the invariants (ξ, ρ, θ), which depend on the principal stresses $(\sigma_1, \sigma_2, \sigma_3)$. Figure from [80].

Article 3 – A physics-informed GAN framework based on model-free data-driven computational mechanics: The article in chapter 4 proposes a novel integration of data-driven computational mechanics within a generative adversarial network framework. This hybrid approach leverages the strengths of both neural networks and data-driven methods. The generator in the GAN is designed to adhere to physical constraints, ensuring consistent and physically viable solutions are generated. Concurrently, the discriminator uses the closest strain-stress data to assess the authenticity of the generator's outputs. This dual mechanism effectively addresses the challenges of sparse data and inconsistency, which are common in standalone data-driven methods and PINNs. This combined approach offers a more robust and versatile framework for simulating and predicting mechanical behavior under various conditions. It demonstrates significant potential in addressing complex scenarios where physics-informed neural networks fail due to the need for consistent physical accuracy. The network structure is further demonstrated in Fig. 1.3.3, which shows how the GAN framework is adapted for model-free data-driven applications.

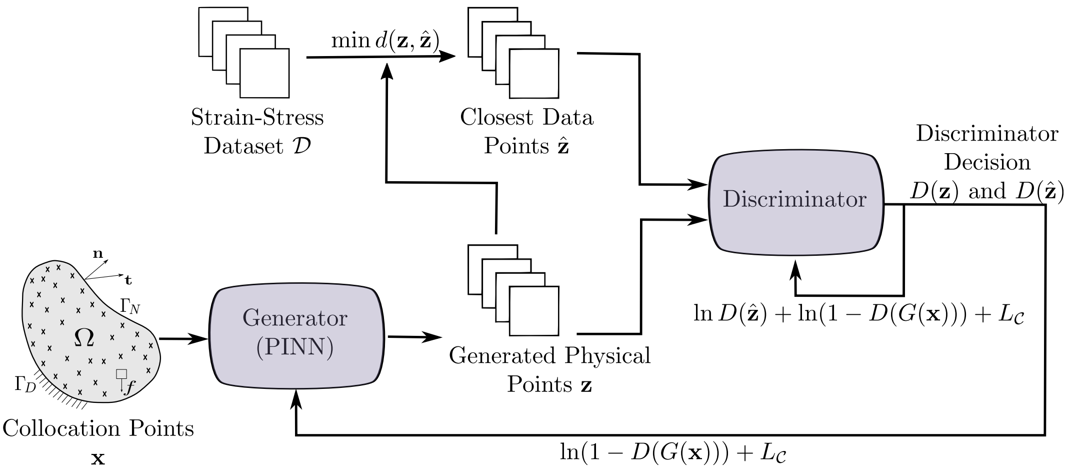

$\min d(\mathbf{z}, \hat{\mathbf{z}})$
Strain-Stress Dataset $\mathcal{D}$
Closest Data Points $\hat{\mathbf{z}}$
Discriminator Decision $D(\mathbf{z})$ and $D(\hat{\mathbf{z}})$
Discriminator
$\mathbf{n}$
$\mathbf{t}$
Γ_N
Ω
Γ_D
f
Generator (PINN)
Generated Physical Points $\mathbf{z}$
$\ln D(\hat{\mathbf{z}}) + \ln(1 - D(G(\mathbf{x}))) + L_{\mathcal{C}}$
Collocation Points $\mathbf{x}$
$\ln(1 - D(G(\mathbf{x}))) + L_{\mathcal{C}}$

Figure 1.3.3: The GAN Framework adapted for model-free data-driven methods, highlighting the integration of physically informed generation and data-driven validation.

2 Article 1: Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules

This article was published as:

CIFTCI, K., & HACKL, K. (2022). Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules. Computational Mechanics, 70(2), 425-435.

Disclosure of the individual authors' contributions to the article:

In model-free data-driven computational mechanics, this article is the culmination of the collaborative endeavors of K. Ciftci and K. Hackl. The idea, introduced by K. Hackl, is the augmentation of tangent space into the strain-stress data set and the classification of the data into subsets of distinct material behaviors. Thus, transition rules map the modeling points to the various subsets. K. Ciftci wrote the manuscript, led the work on the computational parts described, and handled the numerical simulations. K. Hackl developed the mathematical ideas, and K. Ciftci derived the corresponding equations. Both authors have reviewed the paper and agree with its final form for publication.

Abstract: Model-free data-driven computational mechanics replaces phenomenological constitutive functions by numerical simulations based on data sets of representative samples in stress-strain space. The distance of strain and stress pairs from the data set is minimized, subject to equilibrium and compatibility constraints. Although this method operates well for non-linear elastic problems, there are challenges dealing with history-dependent materials, since one and the same point in stress-strain space might correspond to different material behaviour. In recent literature, this issue has been treated by including local histories into the data set. However, there is still the necessity to include models for the evolution of specific internal variables. Thus, a mixed formulation of classical and data-driven modeling is obtained. In the presented approach, the data set is augmented with directions in the tangent space of points in stress-strain space. Moreover, the data set is divided into subsets corresponding to different material behaviour. Based on this classification, transition rules map the modeling points to the various subsets. The approach will be applied to non-linear elasticity and elasto-plasticity with isotropic hardening.

2.1 Introduction

The simulation of boundary-value problems in solid mechanics typically combine two different types of equations; conservation and constitutive laws. The conservation laws are derived from universal principles containing an axiomatic character. Whereas the constitutive laws are formulated through modeling based on experimental observation. Material modeling aims to find these phenomenological models representing the data in the best way possible. Nevertheless, the process of modeling adds error and uncertainty to the solutions, especially in systems with high-dimensional complexity.

One approach to overcome this problem is the usage of machine learning techniques, especially artificial neural networks, to model material behaviour [81, 82, 83]. The network is built directly from experimental data to recognize and learn the underlying non-linear relations between strain and stresses without the construction of an explicit model. The performance of this approach is studied well for many kind of problems including plasticity [84], high-performance material [85] and multiscale analysis [86]. In relation, various neural network architectures have found applications in prediction [87, 88, 89], modeling [90, 91, 92, 93], control and identification design [94, 95] areas of materials science. Despite their good reliability, neural networks rely on hidden layers. Therefore it is unclear on how much each independent variable is influencing the dependent variables, especially for higher-dimensional cases.

The model-free data-driven method by Kirchdoerfer and Ortiz [30] incorporates experimental material data directly into numerical calculations of boundary-value problems. The method is based on a nearest neighbors approach. Particular in continuum mechanics, the optimization problem consists of calculating the closest point in the material data set consistent with the field equations of the problem i.e. compatibility and equilibrium. Therefore the data-driven method provides an alternative formulation of the classical initial-boundary-value problem completely bypassing the empirical material modeling step. For a variety of elasticity problems like non-linear material behaviour [30, 33, 29, 38, 96], dynamics [42], finite strain [37] and material data identification [65] the approach is elaborated and the associated convergence properties are well analyzed. However, problems arise when dealing with history-dependent data as present in inelastic materials, provides one uses nearest neighbor clustering only. Therefore, Eggersmann et. al. [49] include local histories into the

data set, investigating materials with memory, differential materials and history variables. The data-driven paradigm then consists of minimizing the distance between the evolving data set and a time-dependent constraint set. Nonetheless, it is still necessary to resort to additional models for the evolution of internal variables. Thus, a mixed formulation is obtained consisting of a combination of classical and data-driven modeling. In addition to that a variation of the scheme has been proposed by Karapiperis et. al. [58] considering multiscale modeling. The framework emphasizes the parametrization of material history and the optimal sampling of the mechanical state space.
Recently, the data-driven scheme was extended by the tangent space, which improves the learning of the underlying data structure.Ibanez et. al. [44, 45] introduce a method based on a manifold learning approach mapping the data into a lower-dimensional space to use locally linear embedding. A similar second order data-driven scheme, formulated by Eggersmann et. al. [48], uses tensor voting [97] to obtain point-wise tangent space. This enables the search for additional states close to the original data.

This paper presents a new approach by augmenting the tangent space directly into the distance-minimizing data-driven formulation and classify the underlying data structure into subsets corresponding to different material behaviour. Former leads to a much more concise system of equations and the integration of the tangent space enables interpolation in regions of sparse data sampling, whilst ensuring the internal states to cohere with the data set. The data subset classification allows to to deal with loading paths arising in inelasticity avoiding the reliance of models for the evolution of history variables. To operate on the data classifications, transition rules will be defined to map the internal states of the system to the various subsets. As a consequence, the extended data-driven paradigm evaluates the closest point in the transitioned material data subset consistent with the field equations of the problem and additionally closest to the local tangential direction. In the present study, we assume that all needed data is given. Furthermore, the question about data generation and accessibility from experiments remains open. This is a crucial topic that we are going to address in further research.

To provide a general setting, Section 2.2 introduces the basic definitions and derivation of the classical distance-minimizing data-driven computing method. Section 2.3 presents the extension to inelasticity predicated on the extension of the data sets by tangent space information and the classification of the data into subsets corresponding to different material behaviour. Additionally transition rules are defined to map the modeling points to the various data subsets. Section 2.4 demonstrates the performance of the suggested method via numerical examples employing non-linear elasticity and elasto-plasticity with isotropic hardening. At the end, Section 2.5 summarizes the results and gives recommendations concerning future research topics.

2.2 Classical data-driven computing paradigm

In the following the ordinary data-driven computational mechanics method will be summarized for the readers convenience based on the definitions and formulations in [30, 49]. Let $\Omega \subset \mathbb{R}^d$ with $d \in \mathbb{N}$ be a discretized system encountering displacements $\boldsymbol{u} = \{\boldsymbol{u}_i \in \mathbb{R}^{n_i}\}_{i=1}^{n}$ subjected to applied forces $\boldsymbol{f} = \{\boldsymbol{f}_i \in \mathbb{R}^{n_i}\}_{i=1}^{n}$, where $n \in \mathbb{N}$ is the number of nodes and n_i the dimension at node i. The internal state is characterized by strain and stress pairs $\boldsymbol{z}_e = (\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e) \in \mathbb{R}^{2d_e}$ with $d_e \in \mathbb{N}$ being the dimension of stress and strain at material point

$e = 1, \dots, m$, where $m \in \mathbb{N}$ is the number of material points. The internal state of the system is subject to the compatibility and equilibrium condition

$$\boldsymbol{\varepsilon}_e = \boldsymbol{B}_e \boldsymbol{u}_e, \quad \forall e = 1, \dots, m, \tag{2.1}$$

$$\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{\sigma}_e = \boldsymbol{f}. \tag{2.2}$$

In this case w_e is a positive weight and $\boldsymbol{B}_e$ is a strain-displacement matrix. Defining $\boldsymbol{z} = \{(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e)\}_{e=1}^{m}$, the constraints (2.1) and (2.2) define a subspace

$$\mathcal{C} := \left\{ \boldsymbol{z} \in \bigtimes_{e=1}^{m} \mathbb{R}^{2d_e} : \text{(2.1) and (2.2)} \right\}, \tag{2.3}$$

denoted as constraint set with $\times$ being the Cartesian product. Since the set is material-independent, the connectivity between $\boldsymbol{\varepsilon}_e$ and $\boldsymbol{\sigma}_e$ is still missing. Instead of using a functional relationship, the information about the material is given by means of a data set

$$\mathcal{D} := \left\{ \hat{\boldsymbol{z}} \in \bigtimes_{e=1}^{m} \mathcal{D}_e \right\} \quad \text{with} \quad \mathcal{D}_e := \{(\hat{\boldsymbol{\varepsilon}}_i, \hat{\boldsymbol{\sigma}}_i) \in \mathbb{R}^{2d_e}\}_{i=1}^{n_e}, \tag{2.4}$$

where $n_e \in \mathbb{N}$ being the number of local data points; which classically consists of experimental measurements or data achieved from small scale simulations. To define the data-driven problem the local space $\mathbb{R}^{2d_e}$ will be metricized by means of norms

$$\|\boldsymbol{z}_e\|_e := \frac{1}{2} E_e \|\boldsymbol{\varepsilon}_e\|_2^2 + \frac{1}{2} E_e^{-1} \|\boldsymbol{\sigma}_e\|_2^2, \tag{2.5}$$

with numerical scalar $E_e \in \mathbb{R}^+$, typically being of the type of an elastic stiffness, e.g., a representative Young's modulus. One might remark that this metric differs from the metric proposed in [30]. The corresponding local distance function

$$d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e) := \|\boldsymbol{z}_e - \hat{\boldsymbol{z}}_e\|_e \tag{2.6}$$

with $\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e \in \mathbb{R}^{2d_e}$, can be used to define a distance for $\boldsymbol{z}, \hat{\boldsymbol{z}} \in \bigtimes_{e=1}^{m} \mathbb{R}^{2d_e}$ in the global space by

$$d(\boldsymbol{z}, \hat{\boldsymbol{z}}) := \sum_{e=1}^{m} w_e d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e). \tag{2.7}$$

The distance-minimizing data-driven problem, introduced by [30], reads

$$\arg\min_{\hat{\boldsymbol{z}} \in \mathcal{D}} \arg\min_{\boldsymbol{z} \in \mathcal{C}} d(\boldsymbol{z}, \hat{\boldsymbol{z}}) = \arg\min_{\boldsymbol{z} \in \mathcal{C}} \arg\min_{\hat{\boldsymbol{z}} \in \mathcal{D}} d(\boldsymbol{z}, \hat{\boldsymbol{z}}), \tag{2.8}$$

i.e. the aim is to find the closest point consistent with the kinematics and equilibrium laws to a material data set, or equivalently find the point in the data set that is closest to the constraint set. The approach as well as the convergence and well-posedness have been studied on non-linear elastic material behaviour (cf. [30, 29]). In the following the data-driven paradigm will be extended by the tangent space.

2.3 Extension by tangent space

In the following, we will suggest an extension of the data-driven paradigm by including tangent space information in order to deal with inelastic materials. This is a non-trivial task, since the same point in stress-strain space might correspond to different material behavior. Whereas it is proposed in [49] to include local histories into the data set, we will extend the data set by the tangent space information. For this purpose, let us introduce the extended data set

$$\mathcal{D}^{\text{ext}} = \bigtimes_{e=1}^{m} \mathcal{D}_e^{\text{ext}} \quad \text{with} \quad \mathcal{D}_e^{\text{ext}} := \{(\hat{\boldsymbol{z}}_i, \boldsymbol{C}_i) \,|\, \hat{\boldsymbol{z}}_i \in \mathcal{D}_e, \boldsymbol{C}_i \in \mathbb{R}^{d_e \times d_e}\}_{i=1}^{n_e}, \tag{2.9}$$

where $\boldsymbol{C}_i$ represents the total stiffness matrix at $(\hat{\boldsymbol{\varepsilon}}_i, \hat{\boldsymbol{\sigma}}_i)$, including potential inelastic effects. Thus, the actual independent data is given by $((\hat{\boldsymbol{\varepsilon}}_i, \hat{\boldsymbol{\sigma}}_i), \boldsymbol{C}_i)$, i.e. strain, stress and stiffness matrix. We are fully aware, that measuring $\boldsymbol{C}_i$ experimentally might be a formidable task. However, it might very well be possible combining information on nearby strain and stress pairs and employing material symmetry. We plan to elaborate on this in a subsequent paper. For now, we will simply assume $\boldsymbol{C}_i$ to be available.
The tangent space extension allows to operate on the underlying structure of the phase space of strain and stress pairs. In the following, we will introduce a way to incorporate the tangent space directly into the data-driven computing paradigm.

2.3.1 Data-driven formulation

Recalling the distance-minimization problem (2.8), we start by evaluating the data point $(\hat{\boldsymbol{z}}, \boldsymbol{C}) = \{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^{m}$ in the extended data set closest to the constraint set, i.e.

$$\underset{(\hat{\boldsymbol{z}}, \boldsymbol{C}) \in \mathcal{D}^{\text{ext}}}{\arg\min} \; d(\mathcal{C}, \hat{\boldsymbol{z}}). \tag{2.10}$$

Then, each local optimal data point and its corresponding tangent can then be used to define a map $\boldsymbol{y}_e : \mathbb{R}^{d_e} \to \mathbb{R}^{d_e}$ with

$$\boldsymbol{y}_e(\mathrm{x}_e) = \hat{\boldsymbol{\sigma}}_e + \boldsymbol{C}_e(\mathrm{x}_e - \hat{\boldsymbol{\varepsilon}}_e) \qquad \forall e = 1, \dots, m, \tag{2.11}$$

parametrizing the tangent space as subset of the phase space. Thus, the data sets on which the data-driven paradigm operates can be written as

$$\mathcal{D}^{\Delta} = \bigtimes_{e=1}^{m} \mathcal{D}_e^{\Delta} \quad \text{with} \quad \mathcal{D}_e^{\Delta} := \{(\mathrm{x}_e, \boldsymbol{y}_e(\mathrm{x}_e)) \,|\, \mathrm{x}_e \in \mathbb{R}^{d_e}\}. \tag{2.12}$$

This definition allows to incorporate the local tangent spaces directly into the distance-minimization formulation, i.e.

$$\underset{\boldsymbol{z} \in \mathcal{C}}{\arg\min}\, d(\boldsymbol{z}, \mathcal{D}^{\Delta}), \tag{2.13}$$

using the underlying data structure. For this purpose, the remaining step is the determination of the material state $\boldsymbol{z} = \{(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e)\}_{e=1}^{m} \in \mathcal{C}$ closest to the data sets $\mathcal{D}^{\Delta}$. For given optimal data sets $\Delta\hat{\boldsymbol{z}}_e = (\mathrm{x}_e, \boldsymbol{y}_e(\mathrm{x}_e)) \in \mathcal{D}_e^{\Delta}$, e.g. from a previous iteration, the minimization problem (2.13) can then be written as

$$\begin{aligned} &\text{Minimize} \quad \sum_{e=1}^{m} w_e d_e(\boldsymbol{z}_e, \Delta\hat{\boldsymbol{z}}_e) \\ &\text{s.t.} \quad \boldsymbol{\varepsilon}_e = \boldsymbol{B}_e \boldsymbol{u}_e \quad \text{and} \quad \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{\sigma}_e = \boldsymbol{f}. \end{aligned} \tag{2.14}$$

Taking the Cartesian product structure of $\boldsymbol{z} \in \bigtimes_{e=1}^{m} \mathbb{R}^{2d_e}$ into account, it is reasonable to interchange the summation and the minimization. Therefore it is enough to calculate

$$\min_{\boldsymbol{z}_e \in \mathbb{R}^{2d_e}} d_e(\boldsymbol{z}_e, \Delta\hat{\boldsymbol{z}}_e) = \min_{\boldsymbol{z}_e \in \mathbb{R}^{2d_e}} \|\boldsymbol{z}_e - \Delta\hat{\boldsymbol{z}}_e\|_e \qquad \forall e = 1, \ldots, m, \tag{2.15}$$

where we used the definition of the distance function. One can notice that the minimizing state can be found by evaluating the material state $\boldsymbol{z}_e = \Delta\hat{\boldsymbol{z}}_e$. Enforcing the compatibility constraint by expressing the material strains in terms of displacements $\boldsymbol{z}_e = (\boldsymbol{B}_e \boldsymbol{u}_e, \boldsymbol{\sigma}_e)$ it follows

$$\boldsymbol{\varepsilon}_e = \boldsymbol{B}_e \boldsymbol{u}_e = \mathrm{x}_e \qquad \forall e = 1, \ldots, m, \tag{2.16}$$

$$\boldsymbol{\sigma}_e = \boldsymbol{y}_e(\mathrm{x}_e) = \hat{\boldsymbol{\sigma}}_e + \boldsymbol{C}_e(\mathrm{x}_e - \hat{\boldsymbol{\varepsilon}}_e) \qquad \forall e = 1, \ldots, m. \tag{2.17}$$

Substitution of Eq. (2.16) into Eq. (2.17) leads to

$$\boldsymbol{\sigma}_e = \hat{\boldsymbol{\sigma}}_e + \boldsymbol{C}_e(\boldsymbol{B}_e \boldsymbol{u}_e - \hat{\boldsymbol{\varepsilon}}_e) \qquad \forall e = 1, \ldots, m. \tag{2.18}$$

Now using the equilibrium constraint we have

$$\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T (\hat{\boldsymbol{\sigma}}_e + \boldsymbol{C}_e(\boldsymbol{B}_e \boldsymbol{u}_e - \hat{\boldsymbol{\varepsilon}}_e)) = \boldsymbol{f}. \tag{2.19}$$

Finally reordering leads to a standard linear problem given by

$$\left(\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{C}_e \boldsymbol{B}_e\right) \boldsymbol{u} = \boldsymbol{f} - \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T (\hat{\boldsymbol{\sigma}}_e - \boldsymbol{C}_e \hat{\boldsymbol{\varepsilon}}_e). \tag{2.20}$$

Solving the equation system for $\boldsymbol{u}$ and make use of (2.16) and (2.17), we can calculate the closest local material states to the local data sets ensuring the compatibility and equilibrium condition.

Since we assumed given optimal data points, it remains to determine the stress, strain and tangent space pairs $(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)$ in the local data sets $\mathcal{D}_e^{\text{ext}}$ that result in the closest possible satisfaction of compatibility and equilibrium. The determination of the optimal points can be done iteratively. For given data points $\{(\hat{\boldsymbol{z}}_e^k, \boldsymbol{C}_e^k)\}_{e=1}^m$ at iteration k the modeling points $\{\boldsymbol{z}_e^{k+1}\}_{e=1}^m$ are calculated using the data-driven scheme. Next, we calculate the closest local data points in the extended set to the latest modeling points. The iterations are performed until the data assignment remain unchanged or the global distance $d(\boldsymbol{z}, \hat{\boldsymbol{z}})$ is lower than a predefined tolerance, we reached convergence. A visualization of a single algorithmic loading step is given in Fig. 2.3.1 and the detailed extended data-driven scheme is summarized in Algorithm 1.

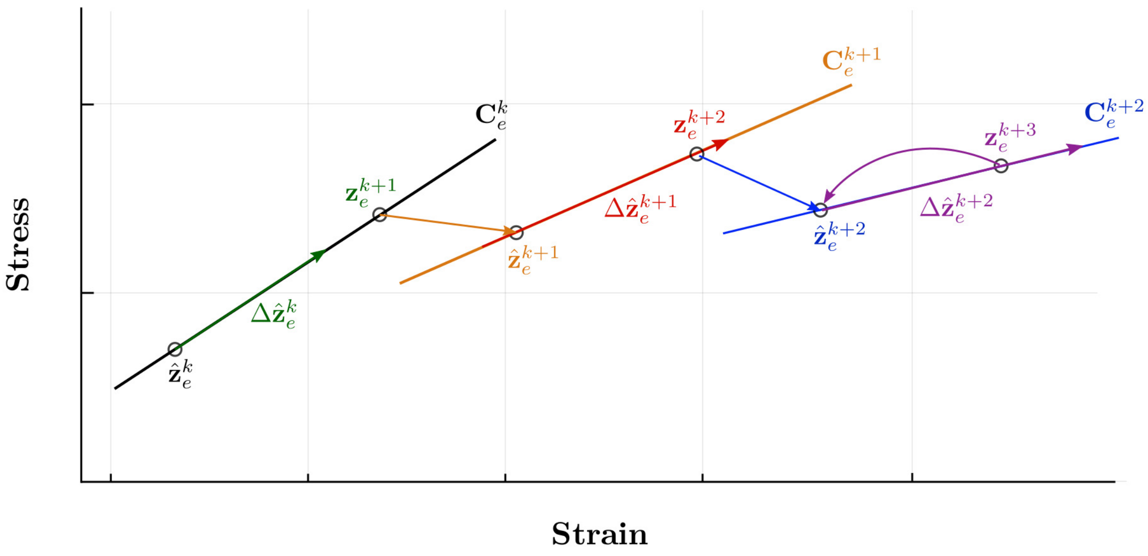


Figure 2.3.1: Visualization of data-driven method extended by tangent space. Modeling points $\boldsymbol{z}_e^{k+1}$ minimize distance to the tangent space associated with data points $\hat{\boldsymbol{z}}_e^k$, respecting compatibility and equilibrium constraints. Data points $\hat{\boldsymbol{z}}_e^{k+1}$ minimize distance to modeling points $\boldsymbol{z}_e^{k+1}$. Iterations are repeated until the local data assignments remain unchanged or the global distance is less than a certain tolerance.

Due to the usage of the tangent-space structure, only a few or even just one iteration are required. This constitutes a considerable increase of efficiency in comparison with the traditional data-driven algorithm introduced in [30]. We realize that the issue about the accessibility of data and its corresponding tangent space is crucial. As mentioned before, we are assuming that all required data is given. The question about data generation and availability will be addressed in further research.

2.3.2 Transition rules

To simulate inelastic material behaviour, the main task is to capture history dependence. This is achieved by associating different tangent spaces to data points with different history. Assuming an underlying data structure, as proposed in [48], the local material data sets $\mathcal{D}_e^{\text{ext}}$ are classified into subsets corresponding to different material behaviour, e.g. elastic and inelastic:

$$\mathcal{D}_e^{\text{ext}} = \dot{\bigcup_p} \mathcal{D}_e^{\text{ext},\,p} \quad \text{with } p = \{\text{elastic},\, \text{inelastic}\}. \tag{2.21}$$

Thus, data points with close or even the same strain and stress values may possess vastly different tangent spaces; in the elastic case essentially determined by the elastic stiffness and in the plastic case by the hardening modulus. It should be emphasized that it is easily possible to distinguish experimentally between elastic and plastic material behaviour. Based on the classification, transition rules map the modeling points to the various subsets.
In the following, a transition mapping is derived for the case of elasto-plasticity with isotropic hardening. The kinematics of elasto-plasticity is governed by a yield condition of the form

$$\sigma_{\text{com}}(\boldsymbol{\sigma}) \leq \sigma_{\text{y}}, \tag{2.22}$$

Algorithm 1 Extended data-driven solver

```
Input: matrices $\{\boldsymbol{B}_e\}_{e=1}^m$, weights $\{w_e\}_{e=1}^m$, load $\boldsymbol{f}$, tolerance $\mathrm{tol}$
Data: data points $\{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^m$, data sets $\{\mathcal{D}_e^{\text{ext}}\}_{e=1}^m$
  function DDSOLVER($\{\mathcal{D}_e^{\text{ext}}\}_{e=1}^m, \{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^m, \boldsymbol{f}$)
    Set iteration $k = 0$
    $\{(\hat{\boldsymbol{z}}_e^k, \boldsymbol{C}^k)\}_{e=1}^m \leftarrow \{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^m$
    while true do
      Solve equation system:
        $$\left(\sum_{e=1}^m w_e \boldsymbol{B}_e^T \boldsymbol{C}_e^k \boldsymbol{B}_e\right) \boldsymbol{u}^{k+1} = \boldsymbol{f}^{k+1} - \sum_{e=1}^m w_e \boldsymbol{B}_e^T (\hat{\boldsymbol{\sigma}}_e^k - \boldsymbol{C}_e^k \hat{\boldsymbol{\varepsilon}}_e^k)$$
      for $e = 1 \to m$ do
        $\boldsymbol{\varepsilon}_e^{k+1} = \boldsymbol{B}_e \boldsymbol{u}^{k+1}$,
        $\boldsymbol{\sigma}_e^{k+1} = \hat{\boldsymbol{\sigma}}_e^k + \boldsymbol{C}_e^k(\boldsymbol{\varepsilon}_e^{k+1} - \hat{\boldsymbol{\varepsilon}}_e^k)$
      end for
      for $e = 1 \to m$ do
        $\min\{d_e(\boldsymbol{z}_e^{k+1}, \hat{\boldsymbol{z}}_e^{k+1}) \mid (\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e) \in \mathcal{D}_e^{\text{ext}}\}$
      end for
      if $d(\boldsymbol{z}^{k+1}, \hat{\boldsymbol{z}}^{k+1}) \leq \mathrm{tol}$ then
        $\{\boldsymbol{z}_e\}_{e=1}^m \leftarrow \{\boldsymbol{z}_e^{k+1}\}_{e=1}^m$
        break
      else
        $k \leftarrow k + 1$
      end if
    end while
    return $\{\boldsymbol{z}_e\}_{e=1}^m$
  end function
```

where $\sigma_{\text{com}}(\boldsymbol{\sigma})$ is a comparison stress dependent on the current stress state, e.g. $\sigma_{\text{com}}(\boldsymbol{\sigma}) = \sqrt{3/2}\,\|\text{dev}\boldsymbol{\sigma}\|$ in the case of von Mises (J_2) plasticity, and σ_{y} denotes the yield stress, a material property depending on the loading history in the case of isotropic hardening. For $\sigma_{\text{com}}(\boldsymbol{\sigma}) < \sigma_{\text{y}}$, we have elastic behaviour, for $\sigma_{\text{com}}(\boldsymbol{\sigma}) = \sigma_{\text{y}}$ plastic behaviour.

Given values of modeling points $\{\boldsymbol{z}_e\}_{e=1}^m$ using the data-driven algorithm 1, the transition mapping for material state $e = 1, \ldots, m$ at time step $t + 1$ can be formulated as:

1. assign local data set $\tilde{\mathcal{D}}_e^{\text{ext}} = \mathcal{D}_e^{\text{ext},\,p}$ by

$$p \equiv \begin{cases} \text{elastic}, & \text{if } \sigma_{\text{com}}(\boldsymbol{\sigma}_e) < \sigma_{\text{y},e} \\ \text{inelastic}, & \text{otherwise.} \end{cases} \tag{2.23}$$

2. if $p \equiv$ inelastic, set new yield stress at $\sigma_{\text{y},e} := \sigma_{\text{com}}(\boldsymbol{\sigma}_e)$;

3. find closest data point $\{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^m$ in data set $\tilde{\mathcal{D}}_e^{\Delta}$ to modeling point $\boldsymbol{z}_e$ by

$$\min\{d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e) \mid (\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e) \in \tilde{\mathcal{D}}_e^{\text{ext}}\}. \tag{2.24}$$

While step 1 maps the modeling points to the corresponding data sets, steps 2 and 3 define a new yield limit and find the closest data point inside these sets for the next loading increment. These formulations give rise to corresponding representational scheme in the context of data-driven inelasticity, which are summarized in Algorithm 2.

Algorithm 2 Data-driven transition rules for inelasticity at time step $t+1$

```
Input: load $\boldsymbol{f}$, yield stresses $\{\sigma_{\mathrm{y},e}\}_{e=1}^{m}$
Data: data points $\{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^{m}$, data subsets $\{(\mathcal{D}_e^{\text{ext, elastic}}, \mathcal{D}_e^{\text{ext, inelastic}})\}_{e=1}^{m}$
  $\{\boldsymbol{z}_e\}_{e=1}^{m}$ = DDSOLVER($\{\tilde{\mathcal{D}}_e^{\text{ext}}\}_{e=1}^{m}, \{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^{m}, \boldsymbol{f}$)
  for $e = 1 \to m$ do
      if $\sigma_{\text{com}}(\boldsymbol{\sigma}_e) < \sigma_{\mathrm{y},e}$ then
          $\tilde{\mathcal{D}}_e^{\text{ext}} \equiv \mathcal{D}_e^{\text{ext, elastic}}$
      else
          $\tilde{\mathcal{D}}_e^{\text{ext}} \equiv \mathcal{D}_e^{\text{ext, inelastic}}$
          $\sigma_{\mathrm{y},e} = \sigma_{\text{com}}(\boldsymbol{\sigma}_e)$
      end if
      $\min\{d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e) \mid (\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e) \in \tilde{\mathcal{D}}_e^{\text{ext}}\}$
  end for
  $t+1 \leftarrow t+2$
```

2.4 Numerical examples

In this section the performance of the presented data-driven solver extended by the tangential space information will be illustrated in two typical benchmark examples considering the stress analysis of non-linear elastic material and an elasto-plastic von Mises material with isotropic hardening. Based on this, we discuss the accuracy and convergence regarding the number of data points. In this scope, we clarify how to add noise to a data point and how to calculate the error between the numerical and the reference solution.

Noise mostly occurs during the collection of data and indicates uncertainties of the measurement. To simulate these uncertainties in our data sets, we add some random noise to each data point. Based on this, adding noise to a tensor $\mathbf{A} \in \mathbb{R}^{d\times d}$ can be done by $\mathbf{A}^{\text{noise}} = \mathbf{A} + n_p(\mathbf{A} \circ \mathbf{\Sigma})$, where $n_p \in [0,1]$ denotes the percentage of noise, $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ is a random tensor with elements $\Sigma_{i,j} \in [-1,1]$ and $\circ$ is the element-wise product.

The error between a data-driven solution $\boldsymbol{z}^k$ and its corresponding reference solution $\boldsymbol{z}^{k,\text{ref}}$ shall be calculated by means of the root-mean-square deviation of strain and stress defined by

$$\text{RMSD}(\boldsymbol{z})^2 = \frac{\sum_{k=0}^{T} \text{Error}(\boldsymbol{z}^k)^2}{T}, \tag{2.25}$$

where $T \in \mathbb{N}$ is the number of total loading steps, $\boldsymbol{z}_e^k = (\boldsymbol{\varepsilon}_e^k, \boldsymbol{\sigma}_e^k)$ the local data-driven states and $\boldsymbol{z}_e^{k,\text{ref}} = (\boldsymbol{\varepsilon}_e^{k,ref}, \boldsymbol{\sigma}_e^{k,\text{ref}})$ the local reference states at step $k \leq T$. The error is given by

$$\text{Error}(\boldsymbol{z}^k)^2 = \frac{\sum_{e=1}^{m} w_e \|\boldsymbol{z}_e^k - \boldsymbol{z}_e^{k,\text{ref}}\|^2}{\sum_{e=1}^{m} w_e \|\boldsymbol{z}_e^{k,\text{ref}}\|^2}, \tag{2.26}$$

with $\|\cdot\|$ given by definition (2.5).

2.4.1 Non-linear elastic cylindrical tube under internal pressure

The first example is the classical benchmark problem considering a non-linear elastic cylindrical tube under internal pressure p. Fig. 2.4.1 illustrates the geometry and the boundary conditions for this particular problem. The tube has thickness $r_2 - r_1$ with inner and outer radii $r_1 = 1\mathrm{m}$ and $r_2 = 2\mathrm{m}$. Two symmetry planes can be identified and therefore the solution domain need only cover a quarter of the geometry, shown by the shaded area. The domain is discretized by quadratic triangles. Modeling the tube as a two dimensional plane-strain problem, the corresponding material parameters of the reference solid used for the reference solution and data sets are Young's modulus $E = 70 \cdot 10^3\,$Pa, Poisson's ratio $\nu = 0.3$ and elasticity tensor

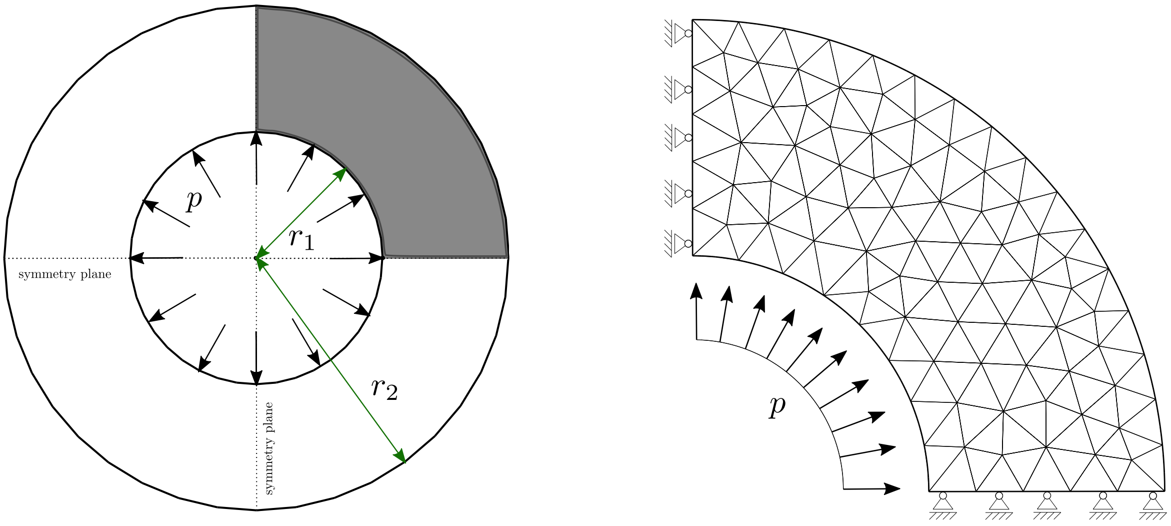


Figure 2.4.1: Discretization and boundary conditions for a cylindrical tube under internal pressure.

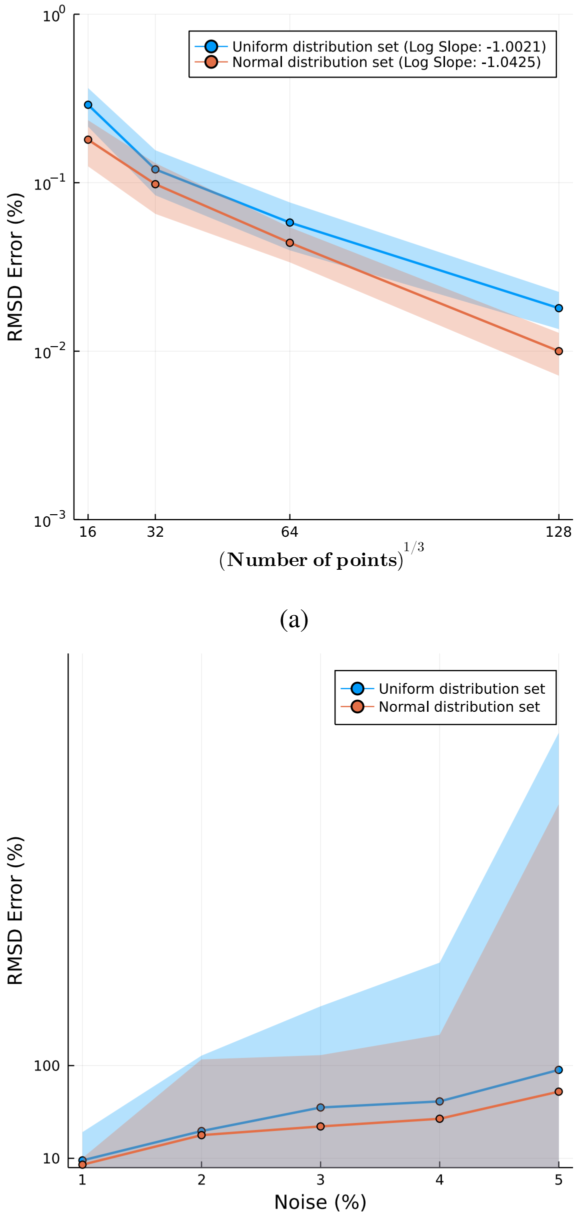
$$\mathbb{C} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{I}, \tag{2.27}$$

where $\mathbf{I}$ is the second-rank identity tensor, $\mathbb{I}$ is the symmetric part of the fourth-rank identity tensor and $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$ are the Lamé constants. The response is computed using a non-linear relation

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}) = \lambda f(\mathrm{tr}(\boldsymbol{\varepsilon}))\mathbf{I} + \mu\boldsymbol{\varepsilon} + \mathbb{C} : \boldsymbol{\varepsilon} \tag{2.28}$$

with $f(x) := c_1 \arctan(c_2 x)$, parameters $c_1 = 3.0 \cdot 10^{-2}$, $c_2 = 1.0 \cdot 10^2$. For the data-driven computation two different types of data distributions are investigated. The first data set is created by a zero-mean normal distribution with a standard deviation of 0.01 and the second data set is created by a uniform distribution within $[-0.02, 0.02]$ for strains in each direction. The corresponding local tangents $\boldsymbol{C}$ are calculated analytically using Eq. (2.28) and endued with some noise. Finally, the simulation of problem in Fig. 2.4.1 is performed by applying a pressure $p(t) = \dfrac{5 \cdot 10^4}{\sqrt{3}} \log\left(\dfrac{r_2}{r_1}\right) \cdot t$ progressively increased with 100 incremental steps using a constant normalized time step of $\Delta t = 1$. Due to the random nature of the data distribution, each simulation returns a different error. To cover a wide spectrum of the errors produced, we run 100 simulations corresponding to independent realizations of both, normal and uniform distribution.

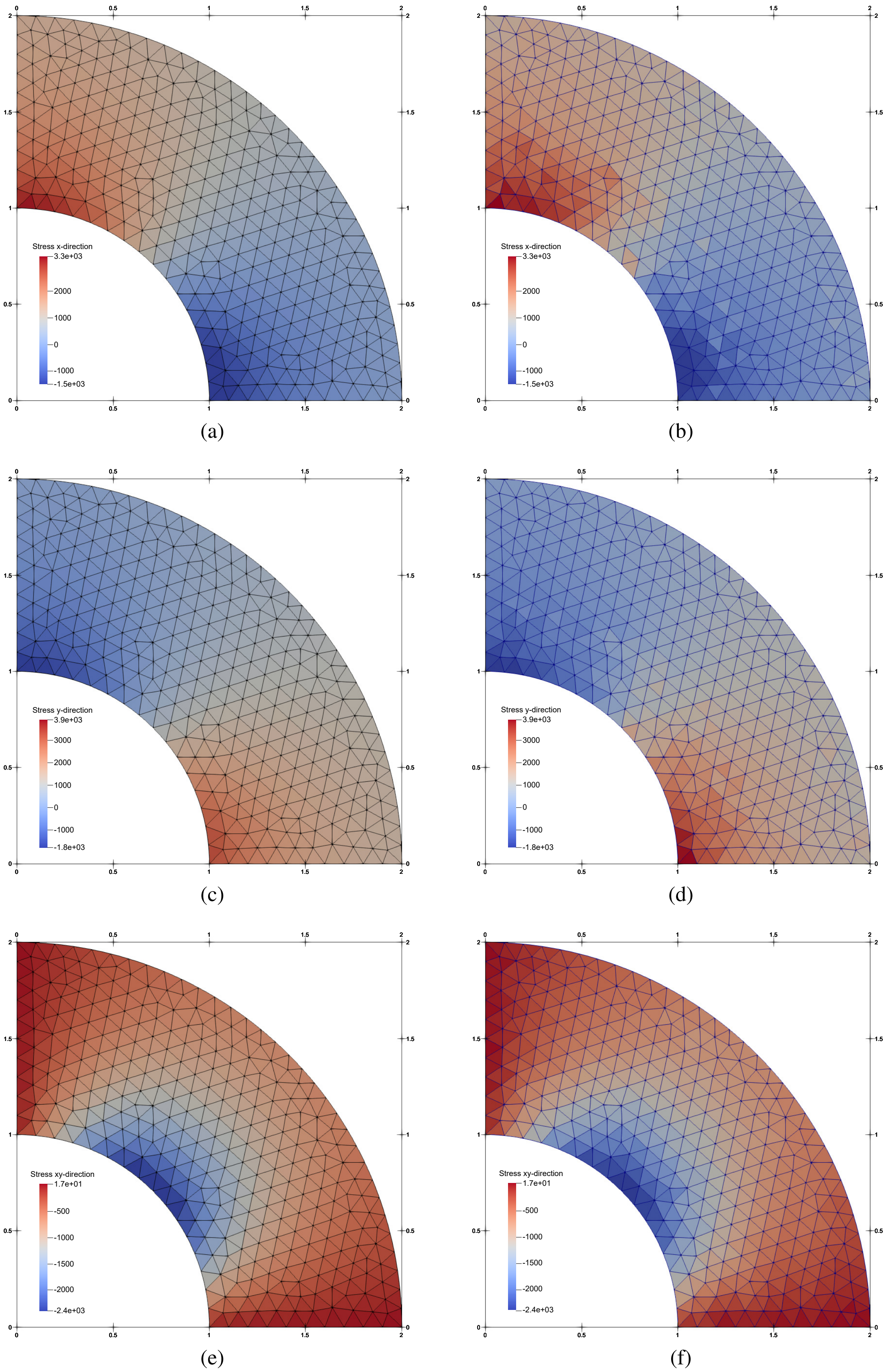
The error plot in Fig. 2.4.2a shows a linear rate of convergence, which corresponds to the data-driven convergence analysis of elastic problems in [30]. Figure 2.4.2b shows the dependence of the error from noising ranging from 1% to 10% of the maximum values of strains and stresses applied to the various data sets. The shaded areas show the spread of the error arising from the different data set realizations used in the independent simulation runs. Note, that apparently both data sets realizations, normal and uniform distribution, yield similar results and convergence performances. Figure 2.4.3 illustrates the occurring stress components σ_{xx}, σ_{yy} and σ_{xy} in the non-linear elastic cylindrical tube under pressure. In addition, it compares the stress distribution produced by the non-linear reference model and the data-driven paradigm based on a normal distribution of size 16^3.



(a)

(b)

Figure 2.4.2: RMSD Error of data-driven solver for normal and uniform distributed data points. (a) Convergence with respect to data size. (b) Dependency of the error on applied noising for a data set of size 16^3 with normal and uniform distribution. The shaded areas show the spread of the error arising from the different data set realizations.



(a) (b)

(c) (d)

(e) (f)

Figure 2.4.3: Comparison of stress components σ_{xx} (a) and (b), σ_{yy} (c) and (d) and σ_{xy} (e) and (f) in $[Pa]$ between the reference model (left) and the data-driven algorithm (right) based on a normal distribution of size 16^3.

2.4.2 Elasto-plastic plate with a circular hole

This example illustrates the performance of the data-driven method extended by transition rules by considering an elasto-plastic von Mises material with isotropic hardening for the boundary value problem in Fig. 2.4.4. The plate with a hole has the dimensions of $\ell = 1\mathrm{m}$, $h = 0.2\,\mathrm{m}$ and $r = 0.05\mathrm{m}$, is clamped at its left edge and subjected to a uniform vertical load q at its right edge. The applied load increases from 0 to $1.8 \cdot 10^7$ Pa, decreases to 0 and then increases again to $2 \cdot 10^7$ Pa, using a constant normalized time step of $\Delta t = 1$.

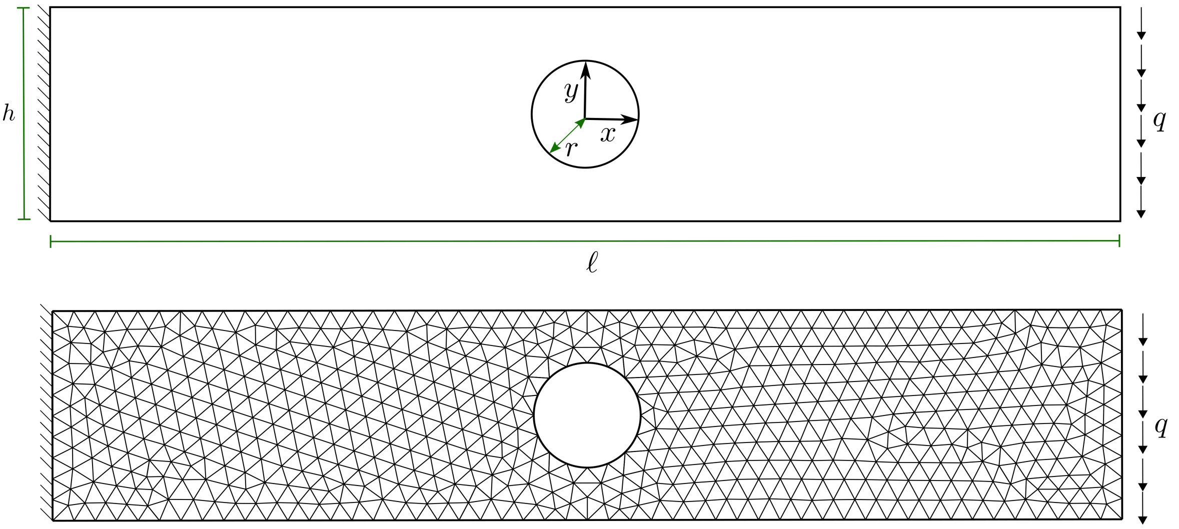


Figure 2.4.4: Discretization and boundary conditions for a rectangular plate with a circular hole under loading.

The material parameters of the reference solid used for the reference solution and data sets are Young's modulus $E = 200 \cdot 10^9$ Pa, Poisson's ratio $\nu = 0.3$, isotropic hardening modulus $H = E/20$, initial yield stress $\sigma_{\mathrm{y}0} = 250 \cdot 10^6$ Pa and elasticity tensor given by

$$\mathbb{C} = (\kappa - \frac{2}{3}G)\mathbf{I} \otimes \mathbf{I} + 2G\mathbb{I}, \tag{2.29}$$

where $\mathbf{I}$ is the second-rank identity tensor, $\mathbb{I}$ is the symmetric part of the fourth-rank identity tensor and $\kappa = \frac{E}{3(1-2\nu)}$ and $G = \frac{E}{2(1+\nu)}$ are the bulk and shear moduli. The response is computed using a J_2-plasticity model based on an iterative return mapping algorithm embedded in a Newton-Raphson global loop restoring equilibrium.

Following [49], a virtual test employing the geometry depicted in Fig. 2.4.5 is used to generate an accurate coverage of suitable local material states and loading paths of various set sizes. As mentioned before, the corresponding tangents are assumed to be given and therefore calculated analytically using the plasticity model. To ensure uncertainties we endure the total stiffness matrices with some noise.
Figure 2.4.6 shows the data-driven solution at the maximum loading magnitude using a data sample containing 10^4 points. The convergence of the maximum displacement to the reference displacement based on a $J2$-plasticity model can be seen in Fig. 2.4.7a. Moreover, Fig. 2.4.7b confirms a linear convergence rate towards the reference solution by increasing the number of data points. For better representation, the convergence of the displacement is shown for only one virtual test. However, the convergence of the error is shown for various virtual tests leading to a deviation visualized by the shaded area.

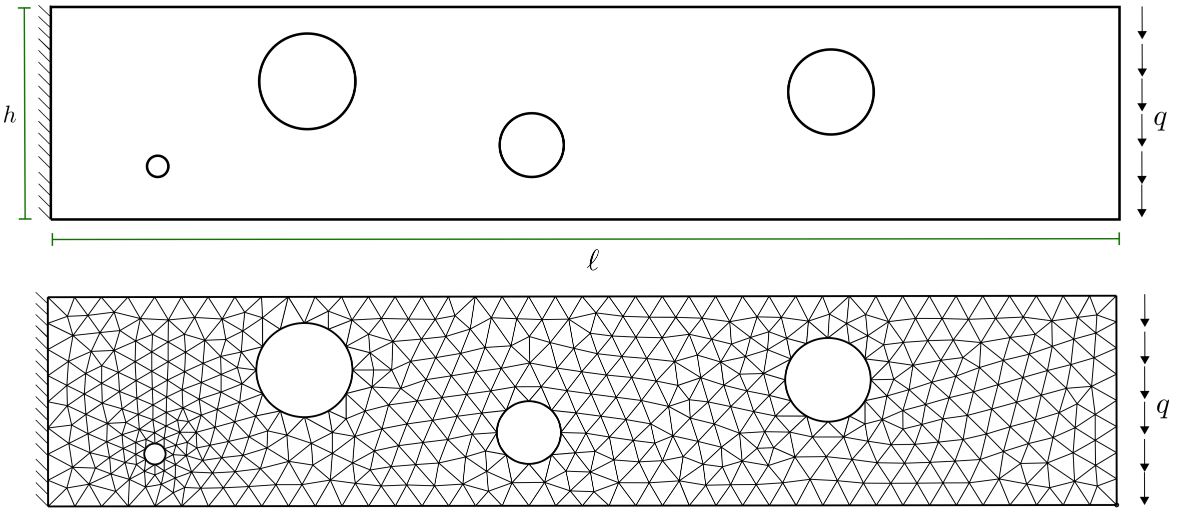
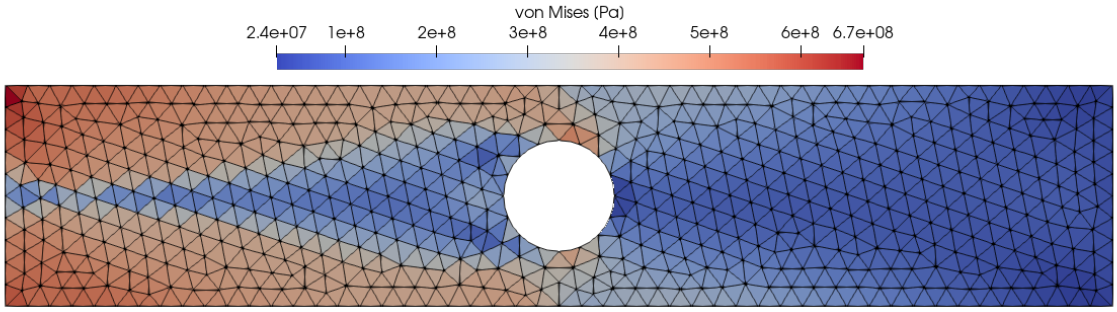
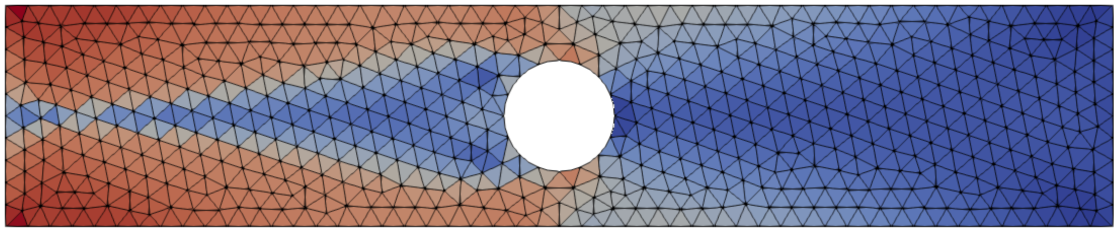


Figure 2.4.5: Geometry and discretization of an example virtual test of a plate with random holes to generate suitable data sets.

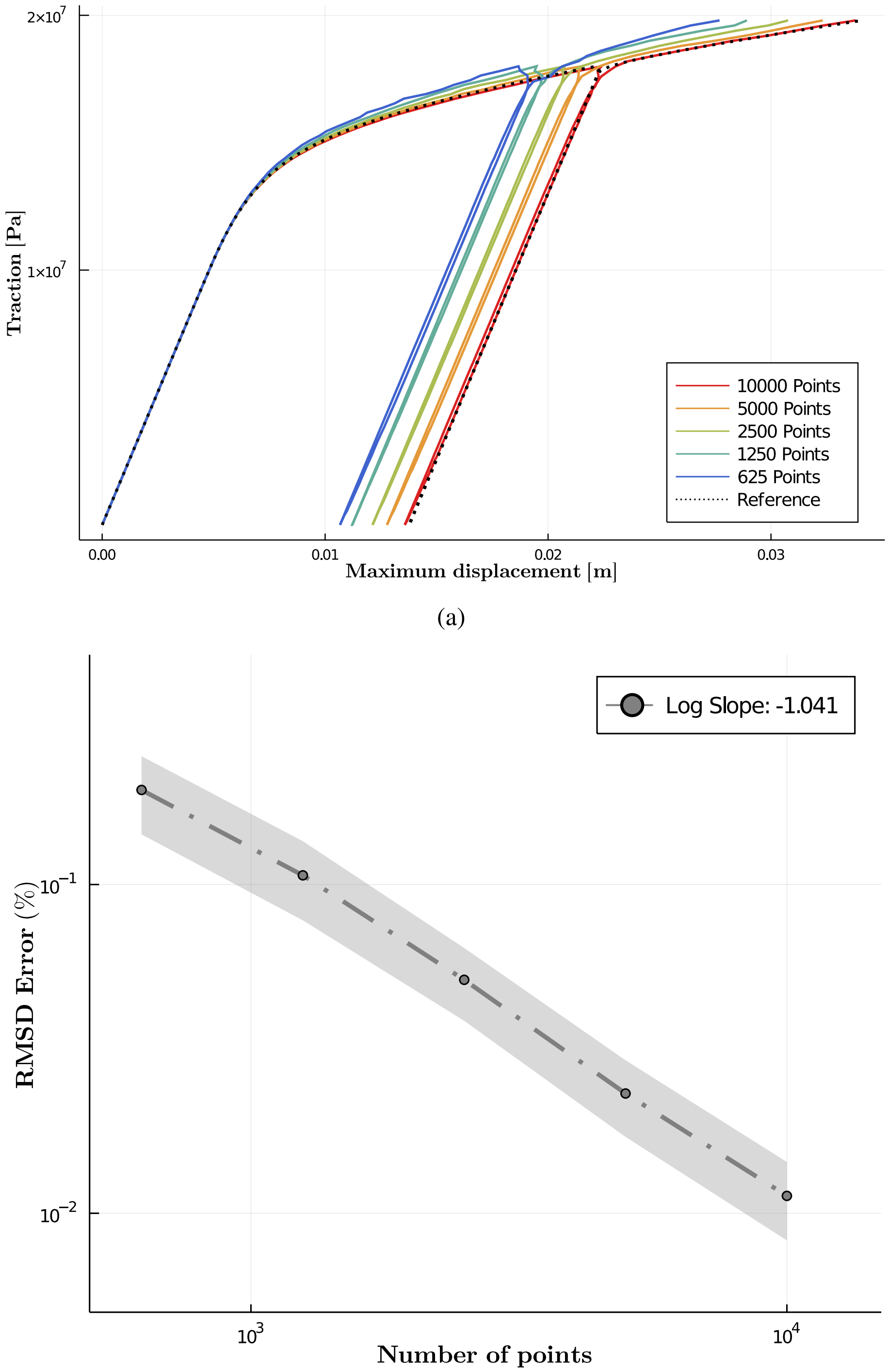


(a) Reference solution



(b) Data-driven solution

Figure 2.4.6: Von Mises stress distribution at maximum loading at each Gaussian integration point using (a) J_2-plasticity model and (b) data-driven algorithm.



(a)

(b)

Figure 2.4.7: Convergence property of the extended data-driven method using transition rules for elasto-plastic material behaviour. (a) Maximum displacement (vertical displacement of lower right vertex versus traction (resultant load of right edge) for different data resolution. (b) RMSD Error for each data resolution. The shaded area shows the deviation of the error arising from different independent virtual tests.

2.5 Conclusions

We present an approach extending the model-free data-driven computing method of problems in elasticity of Kirchdoerfer and Ortiz [30] to inelasticity. The original method uses nearest neighbor clustering and therefore challenges arise dealing with history-dependent data. This issue is treated in this work by extending the formulation by including point-wise tangent spaces and classifying the data structure into subsets corresponding to different material behaviour. Based on the classification, transition rules are defined to map the material point to the classified data subsets, which incorporates with the idea that data points are connected by an underlying structure to each other. Additionally, minimizing the distance to local tangent spaces ensures data point connectivity and enables interpolation in regions lacking information of data. Furthermore, the presented scheme can be easily applied to non-linear elasticity as well, noticing that the resulting system of equations of the minimization problem is reduced, leading to greater efficiency. A numerical example has been presented to demonstrate the application to data-driven inelasticity and its numerical performance.

Generally, it can be concluded that improvements in accuracy of the presented approach increase for larger data sets and it correlates with the convergence analysis of data-driven elasticity. Nevertheless, it should be mentioned that the ensurance of specific quality of the data such as good coverage of material states and loading paths constitutes a critical issue concerning the availability of real experimental data. Another issue concerns the classification of the data into subsets corresponding to material behaviour. This could be done by efficient machine-learning algorithms e.g. spectral or density based clustering. These generalizations of the data-driven paradigm suggest important directions for future research in the area of machine-learning methods, especially physics-informed neural networks. By defining appropriate loss-functions, these networks can not only be trained to satisfy training data but to find optimal solutions for given physics governing equations. Since the data-driven paradigm bypasses the material modeling step but still relies on solving governing equations, a coupled formulation of the model-free data driven and the physics-informed neural network method is conceivable.

3 Article 2: Model-free data-driven inelasticity in Haigh-Westergaard space – a study how to obtain data points from measurements

This article was published as:

CIFTCI, K., & HACKL, K. (2023). Model-free data-driven inelasticity in Haigh-Westergaard space – A study how to obtain data points from measurements. Computer Methods in Applied Mechanics and Engineering, 416, 116352.

Disclosure of the individual authors' contributions to the article:

Within the model-free data-driven computational mechanics paradigm, this research represents a collaborative work by K. Ciftci and K. Hackl. The idea of transforming the data-driven paradigm to the Haigh-Westergaard stress space addressing inelasticity was conceived by K. Hackl. In contrast, K. Ciftci was responsible for the manuscript and implementation of the data-driven solver described in the paper. Additionally, he conducted all the numerical simulations. K. Hackl derived the foundational formulas for the study, and K. Ciftci proved these mathematical formulations to ensure their validity. Both authors have reviewed and endorsed the final version of the manuscript for publication.

Abstract: Model-free data-driven computational mechanics, first proposed by Kirchdoerfer and Ortiz, replaces phenomenological models with numerical simulations based on sample data sets in strain-stress space. Recent literature extended the approach to inelastic problems using structured data sets, tangent space information, and transition rules. From an application perspective, the coverage of qualified data states and calculating the corresponding tangent space is crucial. In this respect, material symmetry significantly helps to reduce the amount of necessary data. This study applies the data-driven paradigm to elasto-plasticity with isotropic hardening. We formulate our approach employing Haigh-Westergaard coordinates, providing information on the underlying material yield surface. Based on this, we use a combined tension-torsion test to cover the knowledge of the yield surface and a single tensile test to calculate the corresponding tangent space. The resulting data-driven method minimizes the distance over the Haigh-Westergaard space augmented with directions in the tangent space subject to compatibility and equilibrium constraints.

3.1 Introduction

In computational mechanics, the simulation of boundary value problems typically combines two different types of equations; conservation and constitutive laws. While conservation laws are derived from universal principles assuming an axiomatic character, constitutive laws are usually acquired by fitting the parameters of a model to given strain-stress data. Examples of such constitutive models can be found in [98]. However, the process of material modeling is often ill-posed and adds uncertainties to the solutions, especially in systems with high-dimensional complexity. The model-free data-driven method, introduced by Kirchdoerfer and Ortiz [30], bypasses the intermediate step of material modeling, incorporating experimental data directly into numerical calculations of boundary-value problems.

The method is elaborated for a variety of applications like non-linear elasticity [30, 33, 29, 38], dynamics [42], finite strain [37] and material data identification [65]. Additional applications can be found in the area of non-local mechanics [56], coupled electro-mechanical [55] and electromagnetic field problems [99, 100], decoupled homogenization schemes [61], and model-driven coupling [54]. An extension of the data-driven scheme has been made by using the tangent space to improve the learning of the underlying data structure. Ibañez et al. [44, 45] suggest a manifold learning approach mapping the data to a lower-dimensional space to use the locally linear embeddings. Eggersmann et al. [48] presented a second-order data-driven approach that uses tensor voting [97] to obtain point-wise tangent spaces enabling the search for additional states close to the original data. For inelastic boundary value problems, Eggersmann et al. [49] include local histories in the data set to investigate materials with memory. Karapiperis et al. [58] have also suggested a variation of the scheme, considering multiscale modeling. In addition, the authors recently developed a method that incorporates the tangent space into the distance-minimizing data-driven formulation and classifies the underlying data structure into subsets according to various material behavior [101]. The framework uses a parametrization of the material history and an optimal sampling of the mechanical system's state space.

Nevertheless, due to possible changes from elastic to plastic behavior as a function of the loading path, it is difficult to deal with data dependent on history as present in inelastic materials, provided one uses nearest neighbor clustering only. Eggersmann et al. [49] overcome this issue by accounting local histories in the data set, investigating three paradigms, i.e., materials with memory regarding the history of deformation, differential materials regarding histories of stress and strain, and history variables. However, it is still necessary to resort to additional models to analyze the evolution of internal variables. The proposed data-driven approach minimizes the distance between the evolving data set and a time-dependent constraint set. The result is a mixed formulation comprising classical and data-driven modeling approaches. Karapiperis et al. [58] have also proposed a variation of the scheme, considering multiscale modeling. The framework uses a parametrization of the material history and an optimal sampling of the mechanical system's state space.

We recently developed a novel strategy that directly incorporates the tangent space into the distance-minimizing data-driven formulation and classifies the underlying data structure into subsets according to different material behavior [101]. The former results in a significantly more compact system of equations and allows for interpolation in sparse data regions guaranteeing that the internal states cohere with the data set. Categorization into data subsetsÂ permits dealing with inelastic loading paths avoiding reliance on models for the evolution of history variables. In addition, we define transition rules mapping the system's internal states to the various subsets to operate on the data categories. As a result, the extended data-driven paradigm locates the closest point in the transitioning material data subset compatible with the problem's field equations and nearest to the local tangential direction. Although the approach works sufficiently well for ideal data, the issue of data accessibility and its accompanying tangent space arises. In particular, data sets of inelastic materials could only be gathered by impractical sample testing encompassing a wide variety of loading directions. Furthermore, using data-driven simulation necessitates the availability of stress-strain pairs encompassing all six components, accompanied by their respective experimental setups. However, the pure application of a data-driven approach becomes unmanageable and impracticable due to this requirement. Consequently, model-free data-driven formulations require a model that extrapolates experimental data in alternative directions.

This paper addresses the data accessibility issue for isotropic elasto-plastic materials. To overcome the reliance on specific directions and quantities of strain-stress data points, we propose a comprehensive data-driven approach that leverages the principal stress space by employing the Haigh-Westergaard coordinate transformation. In regard, the tangent space is characterized by the hardening of the material and the normal on the yield surface. For the former, we use data from a simple tensile test. For the latter, we project the data structure using Haigh-Westergaard coordinates to the octahedral plane in principal stress space. A combined tension-torsion test covers the underlying material yield surface information such that the associated yield function can be determined using a preferred approximation method, e.g., interpolation. The resulting data-driven scheme minimizes the distance to the tensile test data and determines the associated tangent stiffness in the Haigh-Westergaard space, subject to compatibility and equilibrium restrictions. By adopting this methodology, we aim to alleviate the limitations associated with the availability and distribution of data, eventually enhancing the applicability of data-driven modeling in the field of isotropic elasto-plastic materials.

Section 3.2 provides a general setting by introducing the definitions and derivation of the distance-minimizing data-driven computing method, including the enhancement of tangent space and transition rules. Section 3.3 introduces a coordinate transformation to the octahedral plane based on the Haigh-Westergaard coordinates. In addition, we propose an approach to determine the normal on the yield surface and the corresponding material stiffness tangent. Furthermore, we introduce transition rules to map the modeling points to the tangent space. Section 3.4 exhibits the performance of the proposed method using a numerical example involving elasto-plasticity with isotropic hardening. Finally, Section 3.5 summarizes the results and suggests future research subjects.

3.2 Tangent space enhanced data-driven paradigm

The following will summarize the classical data-driven computational mechanics method for the reader's convenience based on the definitions and formulations in [30, 49]. Let $\Omega \subset \mathbb{R}^d$ with $d \in \mathbb{N}$ be a discretized system consisting of $n \in \mathbb{N}$ nodes representing displacements $\boldsymbol{u} \in \mathbb{R}^n$, which are subjected to applied forces $\boldsymbol{f} \in \mathbb{R}^n$ resulting from distributed sources and Neumann boundary conditions. In addition, the system Ω comprises $m \in \mathbb{N}$ material states characterized by strain and stress fields $\boldsymbol{\varepsilon}_e \in \mathbb{R}^{d_e}$ and $\boldsymbol{\sigma}_e \in \mathbb{R}^{d_e}$, with $d_e \in \mathbb{N}$ being the dimension in Voigt notation at material point $e = 1, \dots, m$. The system's internal state is subject to the compatibility and equilibrium conditions

$$\boldsymbol{\varepsilon}_e = \boldsymbol{B}_e \boldsymbol{u}, \quad \forall e = 1, \dots, m, \tag{3.1}$$

$$\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{\sigma}_e = \boldsymbol{f}, \tag{3.2}$$

with volumes associated positive weights $w_e \in \mathbb{R}_+$, discrete gradient operator $\boldsymbol{B}_e \in \mathbb{R}^{d_e \times n}$ and discrete divergence operator $\boldsymbol{B}_e^T \in \mathbb{R}^{n \times d_e}$. Further, we define the set

$$Z := \bigtimes_{e=1}^{m} Z_e \quad \text{with} \quad Z_e := \{(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e) \,|\, \boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e \in \mathbb{R}^{d_e}\}, \tag{3.3}$$

where $Z_e \subseteq \mathbb{R}^{d_e} \times \mathbb{R}^{d_e}$ is the local phase space of material point e, and $Z \subseteq \mathbb{R}^{md_e} \times \mathbb{R}^{md_e}$ is the global phase space of the finite system Ω.

The distance-minimizing data-driven problem, introduced by [30], reads

$$\arg\min_{\hat{\boldsymbol{z}} \in \mathcal{D}} \arg\min_{\boldsymbol{z} \in \mathcal{C}} d(\boldsymbol{z}, \hat{\boldsymbol{z}}) \tag{3.4}$$

where $\mathcal{C} \subset Z$ denotes the constraint set defined by

$$\mathcal{C} := \Big\{ \boldsymbol{z} \in Z : \text{(3.1) and (3.2)} \Big\}; \tag{3.5}$$

containing all states fulfilling compatibility and equilibrium. The set $\mathcal{D} \subset Z$ consists of experimental measurements or results from small-scale simulations and is defined by

$$\mathcal{D} := \bigtimes_{e=1}^{m} \mathcal{D}_e \quad \text{with} \quad \mathcal{D}_e := \{(\hat{\boldsymbol{\varepsilon}}_i, \hat{\boldsymbol{\sigma}}_i) \in Z_e\}_{i=1}^{n_e}, \tag{3.6}$$

$n_e \in \mathbb{N}$ is the number of local data points associated with the integration point e. The distance $d : Z \times Z \to \mathbb{R}$ is defined by

$$d(\boldsymbol{z}, \hat{\boldsymbol{z}}) := \sum_{e=1}^{m} w_e d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e), \tag{3.7}$$

with local distance function $d_e : Z_e \times Z_e \to \mathbb{R}$ defined by

$$d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e) := \|\boldsymbol{z}_e - \hat{\boldsymbol{z}}_e\|_e \tag{3.8}$$

metricized by the norm

$$\|\boldsymbol{z}_e\|_e := \frac{1}{2} E_e \|\boldsymbol{\varepsilon}_e\|_2^2 + \frac{1}{2} E_e^{-1} \|\boldsymbol{\sigma}_e\|_2^2, \tag{3.9}$$

where $E_e \in \mathbb{R}^+$ is a numerical scalar typically being of the type of elastic stiffness.

Thus, the data-driven method aims to find the closest point $\boldsymbol{z}$ in the constraint set $\mathcal{C}$ to $\hat{\boldsymbol{z}}$ in the material data set $\mathcal{D}$, or equivalently find the point in the data set that is closest to the constraint set.

3.2.1 Tangent space, structured data sets and transition rules

To deal with inelastic materials, we extended the classical data-driven paradigm (3.4) by tangent space information, structured data sets, and transition rules [101]. The phase space collects a physical system's possible strain-stress states that a material can experience under certain conditions. The tangent space extension enables us to operate on the underlying structure of this phase space, which allows us to analyze the system's behavior in a neighborhood of a particular strain-stress point. For this purpose, we recall the definition of the extended data set

$$\mathcal{D}^{\text{ext}} = \bigtimes_{e=1}^{m} \mathcal{D}_e^{\text{ext}} \quad \text{with} \quad \mathcal{D}_e^{\text{ext}} := \{(\hat{\boldsymbol{z}}_i, \boldsymbol{C}_i) \,|\, \hat{\boldsymbol{z}}_i \in \mathcal{D}_e, \boldsymbol{C}_i \in \mathbb{R}^{d_e \times d_e}_{\text{sym},+}\}_{i=1}^{n_e}, \tag{3.10}$$

where $\boldsymbol{C}_i$ represents the symmetric positive definite stiffness matrix at $\hat{\boldsymbol{z}}_i = (\hat{\boldsymbol{\varepsilon}}_i, \hat{\boldsymbol{\sigma}}_i)$, including possible inelastic effects. Incorporating the tangent space directly into the data-driven computing method reads

$$\underset{(\hat{\boldsymbol{z}}, \boldsymbol{C}) \in \mathcal{D}^{\text{ext}}}{\arg\min} \; \underset{\boldsymbol{z} \in \mathcal{C}}{\arg\min} \, d(\boldsymbol{z}, \hat{\boldsymbol{z}}), \tag{3.11}$$

with data points $(\hat{\boldsymbol{z}}, \boldsymbol{C}) = \{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^m$. Based on [30], we determine the optimal points iteratively using a fixed-point iteration expressed by

$$(\hat{\boldsymbol{z}}^{k+1}, \boldsymbol{C}^{k+1}) = P_{\mathcal{D}}(P_{\mathcal{C}}(\hat{\boldsymbol{z}}^k, \boldsymbol{C}^k)), \tag{3.12}$$

where $k \in \mathbb{N}$ denotes the current iteration.

The first mapping $P_{\mathcal{C}} : \mathcal{D}^{\text{ext}} \to \mathcal{C}$ projects a data state $(\hat{\boldsymbol{z}}^k, \boldsymbol{C}^k) \in \mathcal{D}^{\text{ext}}$ to the closest point in the constraint set $\boldsymbol{z}^k \in \mathcal{C}$. For fixed data points $\{(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)\}_{e=1}^m$, e.g., from a previous iteration, the projection is performed by solving the linear equation system [101]

$$\left(\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{C}_e \boldsymbol{B}_e \right) \boldsymbol{u} = \boldsymbol{f} - \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T (\hat{\boldsymbol{\sigma}}_e - \boldsymbol{C}_e \hat{\boldsymbol{\varepsilon}}_e), \tag{3.13}$$

and computing the corresponding strain and stress values by

$$\boldsymbol{\varepsilon}_e = \boldsymbol{B}_e \boldsymbol{u} \qquad \forall e = 1, \dots, m, \tag{3.14}$$

$$\boldsymbol{\sigma}_e = \hat{\boldsymbol{\sigma}}_e + \boldsymbol{C}_e(\boldsymbol{\varepsilon}_e - \hat{\boldsymbol{\varepsilon}}_e) \qquad \forall e = 1, \dots, m. \tag{3.15}$$

The second projection $P_{\mathcal{D}} : \mathcal{C} \to \mathcal{D}^{\text{ext}}$ finds the closest state in the data set to the previously calculated state in the constraint set. We associate different tangent spaces to data points with different histories. The local material data sets $\mathcal{D}_e^{\text{ext}}$ are classified into subsets corresponding to different material behavior, e.g., elastic and inelastic:

$$\mathcal{D}_e^{\text{ext}} = \dot{\bigcup_p} \mathcal{D}_e^{\text{ext},p} \quad \text{with } p = \{\text{elastic, inelastic}\}. \tag{3.16}$$

Based on the classification, transition rules map the modeling points to the various subsets. Thus, the closest point projection $P_{\mathcal{D}}$ is done by minimizing the local distances d_e for the material states $\boldsymbol{z}_e^k$ subject to data subset $\mathcal{D}_e^{\text{ext},p}$. In other words, a nearest-neighbor problem has to be solved to find the data point $(\hat{\boldsymbol{z}}_e^{k+1}, \boldsymbol{C}_e^{k+1}) \in \mathcal{D}_e^{\text{ext},p}$ closest to $\boldsymbol{z}_e^{k+1}$ regarding the metric (3.7).

In [101], we derived such a projection for the case of elasto-plasticity with isotropic hardening. A yield condition governs the kinetics of elasto-plasticity by

$$\sigma_{\text{com}}(\boldsymbol{\sigma}) \leq \sigma_{\text{y}}, \tag{3.17}$$

where $\sigma_{\text{com}}(\boldsymbol{\sigma})$ is comparison stress dependent on the current stress state and σ_{y} denotes the yield stress, a material property depending on the loading history in the case of isotropic hardening. For fixed modeling points $\{\boldsymbol{z}_e\}_{e=1}^m$ e.g. achieved from the linear equation system (3.13), the mapping $P_{\mathcal{D}}$ can be performed for material state $e = 1, \dots, m$ by:

1. check yield condition and assign index

$$p = \begin{cases} \text{elastic}, & \text{if } \sigma_{\text{com}}(\boldsymbol{\sigma}_e) < \alpha_{\text{y,e}} \\ \text{inelastic}, & \text{otherwise;} \end{cases} \tag{3.18}$$

2. if $p \equiv$ inelastic, set new yield stress

$$\sigma_{\text{y,e}} \equiv \sigma_{\text{com}}(\boldsymbol{\sigma}_e); \tag{3.19}$$

3. find closest data point $(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)$ to modeling point $\boldsymbol{z}_e$ by

$$\min\{d_e(\boldsymbol{z}_e, \hat{\boldsymbol{z}}_e) \,|\, (\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e) \in \mathcal{D}_e^{\text{ext},p}\}. \tag{3.20}$$

The first step maps the modeling points to the corresponding data sets; steps 2 and 3 define a new yield limit and find the closest data point inside these sets for the next loading increment.

Studying the data-driven approach for ideal data, we realize that the issue of the accessibility of data and its corresponding tangent space is crucial. While 100 data points are sufficient in the $1D$ case, 100^3 are needed for the $2D$ and 100^6 for the $3D$ case [48]. Considering the latter case, the local data sets consist of strain and stress pairs $(\hat{\boldsymbol{\varepsilon}}_i, \hat{\boldsymbol{\sigma}}_i)$ with 12 independent components. By considering symmetry, the corresponding tangents $\boldsymbol{C}_i$ have 21 independent components. This fact raises the question of obtaining suitable data sets from measurements to cover the phase space and determine the consistent tangent space. Especially data sets for the simulation of inelastic material behavior could only be obtained by impracticable sample tests covering a variety of loading paths. This study demonstrates the suitability of data sets obtained from combined tension-torsion and single tension tests for the data-driven modeling of elasto-plasticity. It should be noted that the proposed approach primarily focuses on isotropic hardening. Nevertheless, it is worth mentioning that the framework can be extended to enclose a broader range of symmetric hardening behaviors.

3.3 Data-driven paradigm in octahedral plane

In order to address the concern regarding the impracticable data sets, we propose the introduction of the octahedral plane within the Haigh-Westergaard coordinate system. This transformation modifies the mapping function, denoted as $P_{\mathcal{D}}$, in the principal stress space, resulting in a reduction of the problem to only three components, specifically $(\sigma_1, \sigma_2, \sigma_3)$. The tangent stiffness $\boldsymbol{C}$ is determined by the normal vector to the yield surface. However, the challenge lies in accurately representing the yield surface itself. For this, we exploit the symmetry inherent in the yield surface to construct the surface by utilizing combined tension-torsion data sets from the symmetric portion. Consequently, an analytical expression can be derived depending on the tangent stiffness hardening variable. The hardening variable can be implicitly obtained using strain-stress increment data from a simple tension test. The data set is subsequently enriched with the aforementioned hardening variable. In order to determine the optimal strain-stress increment, we find the closest hardening variable within the tensile data set to the hardening variable of the material state obtained from the yield surface in Haigh-Westergaard coordinates is identified. Since each material state satisfying the tangent relation in the principal stress space lies on the yield surface, which may evolve during hardening, we utilize such states as new data points in the mapping function $P_{\mathcal{D}}$ to obtain the next material state. In this section, we suggest a way to obtain data from measurements insofar as we have to modify the data-driven fixed-point method's projection $P_{\mathcal{D}}$.

3.3.1 Preliminaries for tangents of isotropic elasto-plastic bodies

We start by introducing the constitutive relation for isotropic elasto-plastic materials

$$\boldsymbol{\sigma} = \lambda \operatorname{tr}(\boldsymbol{\varepsilon})\mathbf{I} + 2\mu(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p), \tag{3.21}$$

with strain $\boldsymbol{\varepsilon}$, stress $\boldsymbol{\sigma}$, plastic strain $\boldsymbol{\varepsilon}^p$, Lamé constants λ, μ and second-rank identity tensor $\mathbf{I}$. We employ strain and stress in means of tensors instead of Voigt form, i.e., $(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d}$. Taking the derivative of Eq. (3.21) and making use of plasticity theory, it follows

$$\dot{\boldsymbol{\sigma}} = \lambda \operatorname{tr}(\dot{\boldsymbol{\varepsilon}})\mathbf{I} + 2\mu(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p) = \boldsymbol{C} : \dot{\boldsymbol{\varepsilon}}, \tag{3.22}$$

with tangent operator

$$\boldsymbol{C} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu\mathbb{I} - \gamma \boldsymbol{N} \otimes \boldsymbol{N}. \tag{3.23}$$

In this context, $\mathbb{I}$ is the symmetric part of the fourth-rank identity tensor, $\gamma \in \mathbb{R}_{\geq 0}$ is a parameter depending on the hardening and $\boldsymbol{N}$ is the normal given by

$$\boldsymbol{N} = \left\| \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \right\|^{-1} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}}, \tag{3.24}$$

with Φ being a continuously differentiable yield function. The LamÃ© constants can be calculated using a simple tension test. Alternatively we can decompose the tangent given in Eq. (3.23) by $\boldsymbol{C} = \boldsymbol{C}^{\mathrm{el}} - \boldsymbol{C}^{\mathrm{pl}}$ with

$$\boldsymbol{C}^{\mathrm{el}} := \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{I} \quad \text{and} \quad \boldsymbol{C}^{\mathrm{pl}} := \gamma \boldsymbol{N} \otimes \boldsymbol{N}. \tag{3.25}$$

The matrix $\boldsymbol{C}^{\mathrm{el}}$ can be calculated using a principal component analysis applied to the elastic part of data [73]. Thus, the main task to obtain tangent $\boldsymbol{C}$ at a fixed point $(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ is the computation of $\boldsymbol{C}^{\mathrm{pl}}$ depending only on normal $\boldsymbol{N}$ and parameter $\gamma \in \mathbb{R}_{\geq 0}$.

3.3.2 Normal vector in the octahedral plane based on Haigh-Westergaard coordinates

The normal $\boldsymbol{N}$ on the yield surface Φ at a fixed point is orthogonal to the corresponding tangent vector at this point, which can be obtained by differentiating the corresponding position vector on the surface. Instead of performing the calculations in the Cartesian coordinate system, we transform the position vector to a curvilinear system, i.e., principal stress space $(\sigma_1, \sigma_2, \sigma_3)$, where each component of the vector is expressed by cylindrical coordinates determining the surface. For this purpose, we introduce the Haigh-Westergaard coordinates (ξ, ρ, θ) describing a cylindrical coordinate system within principal stress space. Coordinate ξ is the projection on the vector $(1, 1, 1)$ of the hydrostatic axis, and (ρ, θ) are polar coordinates in the deviatoric plane that is orthogonal to the hydrostatic axis [102, 103]. The coordinates can be computed using invariants of the stress tensor $\boldsymbol{\sigma}$ and its deviator $\boldsymbol{s}$ defined as

$$J_1 = \mathrm{tr}(\boldsymbol{\sigma}), \tag{3.26}$$

$$J_2 = \frac{1}{2}\left[\mathrm{tr}(\boldsymbol{\sigma}^2) - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})^2\right] = \frac{1}{2}\mathrm{tr}(\boldsymbol{s} \cdot \boldsymbol{s}), \tag{3.27}$$

$$J_3 = \det(\boldsymbol{s}). \tag{3.28}$$

Based on this, the Haigh-Westergaard coordinates (ξ, ρ, θ) can be obtained by

$$\xi = \frac{J_1}{\sqrt{3}} = \frac{\sigma_1 + \sigma_2 + \sigma_3}{\sqrt{3}}, \tag{3.29}$$

$$\rho = \sqrt{2J_2} = \frac{1}{3}\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}, \tag{3.30}$$

$$\begin{aligned} \theta &= \frac{1}{3} \arccos\left(\frac{3\sqrt{3}}{2} J_3 J_2^{-3/2}\right) \\ &= \arccos\left(\frac{2\sigma_1 - \sigma_2 - \sigma_3}{\sqrt{2}\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}}\right), \end{aligned} \tag{3.31}$$

with $\theta \in [0, \frac{\pi}{3}]$. A point $\boldsymbol{\sigma}$ can then be expressed in terms of the coordinates (ξ, ρ, θ) as

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = \frac{\xi}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \sqrt{\frac{2}{3}} \rho \begin{pmatrix} \cos(\theta) \\ \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) \end{pmatrix}, \tag{3.32}$$

with principal stresses $\sigma_1 \geq \sigma_2 \geq \sigma_3$. For $\xi \equiv 0$, the resulting plane, known as the deviatoric or octahedral plane, is a subspace of Haigh-Westergaard space given by (ρ, θ). An illustration of the coordinates and the resulting plane is given in Fig. 3.3.1.

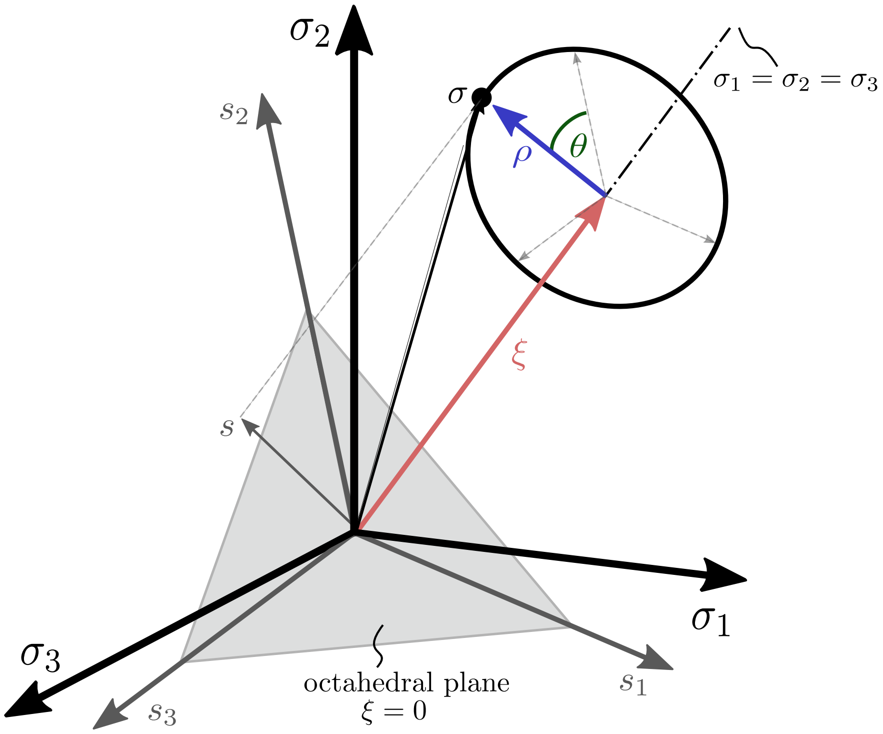


Figure 3.3.1: Visualisation of a stress tensor $\boldsymbol{\sigma}$ and its deviatoric part in the Haigh-Westergaard stress space.

In addition, the intersection of the yield surface with the octahedral plane is given by

$$\rho(\theta) = \alpha\Phi(\theta), \tag{3.33}$$

with $\alpha \in \mathbb{R}_{\geq 1}$ describing isotropic hardening. In particular, for $\alpha = 1$, the equation (3.33) represents the initial yield surface. As mentioned before, the normal to this surface at a fixed point $\boldsymbol{\sigma}$ is perpendicular to any tangent vector at this point (see Fig. 3.3.2). Regarding the

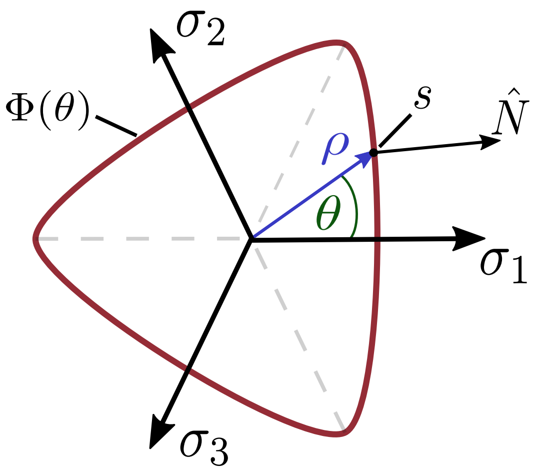


Figure 3.3.2: Schematic illustration of a normal vector $\hat{\boldsymbol{N}}$ at a random point $\boldsymbol{s}$ of a parametrized yield surface Φ in Haigh-Westergaard coordinates

Haigh-Westergaard space, the tangent space is a 2-dimensional plane whose basis consists of two tangent vectors given by

$$\mathbf{t}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{t}_2 = \frac{\partial \boldsymbol{s}}{\partial \theta}, \tag{3.34}$$

where $\boldsymbol{s}$ is the deviatoric part of point $\boldsymbol{\sigma}$ defined in Haigh-Westergaard coordinates as

$$\boldsymbol{s} = \sqrt{\frac{2}{3}}\rho(\theta) \begin{pmatrix} \cos(\theta) \\ \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) \end{pmatrix}. \tag{3.35}$$

The derivative of $\boldsymbol{s}$ with respect to θ is given by

$$\frac{\partial \boldsymbol{s}}{\partial \theta} = \sqrt{\frac{2}{3}}\rho(\theta) \begin{pmatrix} -\sin(\theta) \\ \cos\left(\frac{\pi}{6} - \theta\right) \\ -\cos\left(\frac{\pi}{6} + \theta\right) \end{pmatrix} + \sqrt{\frac{2}{3}}\rho'(\theta) \begin{pmatrix} \cos(\theta) \\ \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) \end{pmatrix}. \tag{3.36}$$

Based on Szeptyński [104], the normal vector can then be calculated with the tangential vectors given in Eq. (3.34) by

$$\hat{\boldsymbol{N}} = -\frac{\mathbf{t}_1 \times \mathbf{t}_2}{\|\mathbf{t}_1 \times \mathbf{t}_2\|_2}, \tag{3.37}$$

where $\|\cdot\|_2$ denotes the well-known Euclidean norm. Substitution of Eq. (3.36) into Eq. (3.34) and using trigonometric identities yields for the cross product of the tangent vectors

$$\begin{aligned} \mathbf{t}_1 \times \mathbf{t}_2 =& \sqrt{\frac{2}{3}}\rho(\theta) \begin{pmatrix} \cos\left(\frac{\pi}{6} - \theta\right) + \cos\left(\frac{\pi}{6} + \theta\right) \\ \sin(\theta) - \cos\left(\frac{\pi}{6} + \theta\right) \\ -\sin(\theta) - \cos\left(\frac{\pi}{6} - \theta\right) \end{pmatrix} \\ &+ \sqrt{\frac{2}{3}}\rho'(\theta) \begin{pmatrix} \sin\left(\frac{\pi}{6} + \theta\right) - \sin\left(\frac{\pi}{6} - \theta\right) \\ -\cos(\theta) - \sin\left(\frac{\pi}{6} + \theta\right) \\ \cos(\theta) + \sin\left(\frac{\pi}{6} - \theta\right) \end{pmatrix} \end{aligned} \tag{3.38}$$

$$=\sqrt{2}\rho(\theta) \begin{pmatrix} \cos(\theta) \\ -\sin\left(\frac{\pi}{6} - \theta\right) \\ -\sin\left(\frac{\pi}{6} + \theta\right) \end{pmatrix} + \sqrt{2}\rho'(\theta) \begin{pmatrix} \sin(\theta) \\ -\cos\left(\frac{\pi}{6} - \theta\right) \\ \cos\left(\frac{\pi}{6} + \theta\right) \end{pmatrix} \tag{3.39}$$

$$=\sqrt{3}\boldsymbol{s} + \sqrt{2}\rho'(\theta) \begin{pmatrix} \sin(\theta) \\ -\cos\left(\frac{\pi}{6} - \theta\right) \\ \cos\left(\frac{\pi}{6} + \theta\right) \end{pmatrix}. \tag{3.40}$$

Applying the definition of Euclidean norm to Eq. (3.40) and making use of trigonometric identities, it follows

$$\begin{align}
\|\mathbf{t}_1 \times \mathbf{t}_2\|_2^2 = & \left\| \sqrt{3}\boldsymbol{s} + \sqrt{2}\rho'(\theta) \begin{pmatrix} \sin(\theta) \\ -\cos\left(\frac{\pi}{6} - \theta\right) \\ \cos\left(\frac{\pi}{6} + \theta\right) \end{pmatrix} \right\|_2^2 \tag{3.41} \\
= & \left(\sqrt{2}\rho(\theta)\cos(\theta) + \sqrt{2}\rho'(\theta)\sin(\theta) \right)^2 \tag{3.42} \\
& + \left(-\sqrt{2}\rho(\theta)\sin\left(\frac{\pi}{6} - \theta\right) - \sqrt{2}\rho'(\theta)\cos\left(\frac{\pi}{6} - \theta\right) \right)^2 \nonumber \\
& + \left(-\sqrt{2}\rho(\theta)\sin\left(\frac{\pi}{6} - \theta\right) - \sqrt{2}\rho'(\theta)\cos\left(\frac{\pi}{6} - \theta\right) \right)^2 \nonumber \\
= & 2\rho(\theta)^2 \left(\cos(\theta)^2 + \sin\left(\frac{\pi}{6} - \theta\right)^2 + \sin\left(\frac{\pi}{6} + \theta\right)^2 \right) \tag{3.43} \\
& + 2\rho'(\theta)^2 \left(\sin(\theta)^2 + \cos\left(\frac{\pi}{6} - \theta\right)^2 + \cos\left(\frac{\pi}{6} + \theta\right)^2 \right) \nonumber \\
& + 2\rho(\theta)\rho'(\theta) \left(2\cos(\theta)\sin(\theta) + \cos\left(\frac{\pi}{6} + 2\theta\right) - \cos\left(\frac{\pi}{6} - 2\theta\right) \right) \nonumber \\
= & 2\rho(\theta)^2 \left(\cos(\theta)^2 + \frac{3}{2}\sin(\theta)^2 + \frac{1}{2}\cos(\theta)^2 \right) \tag{3.44} \\
& + 2\rho'(\theta)^2 \left(\sin(\theta)^2 + \frac{1}{2}\sin(\theta)^2 + \frac{3}{2}\cos(\theta)^2 \right) \nonumber \\
& + 2\rho(\theta)\rho'(\theta) \left(2\cos(\theta)\sin(\theta) - 2\cos(\theta)\sin(\theta) \right) \nonumber \\
& = 3\left(\rho(\theta)^2 + \rho'(\theta)^2 \right). \tag{3.45}
\end{align}$$

The normal vector in principal stress space can then be expressed in Haigh-Westergaard coordinates by substituting Eq. (3.40) and Eq. (3.45) into Eq. (3.37) yielding

$$\hat{\boldsymbol{N}}(\theta) = \frac{1}{\sqrt{\rho(\theta)^2 + \rho'(\theta)^2}} \left[\boldsymbol{s} + \sqrt{\frac{2}{3}}\rho'(\theta) \begin{pmatrix} \sin(\theta) \\ -\cos\left(\frac{\pi}{6} - \theta\right) \\ \cos\left(\frac{\pi}{6} + \theta\right) \end{pmatrix} \right]. \tag{3.46}$$

The normal $\hat{\boldsymbol{N}}$ can be transformed back to Cartesian coordinate system by $\boldsymbol{N} = \mathbf{T} \cdot \operatorname{diag}(\hat{\boldsymbol{N}}) \cdot \mathbf{T}^{-1}$, where $\mathbf{T} \in \mathbb{R}^{3\times 3}$ consists of eigenvectors corresponding to the eigenvalues $\sigma_1 \geq \sigma_2 \geq \sigma_3$ of $\boldsymbol{\sigma}$ and satisfies the transformation to principal stress space $\boldsymbol{\sigma} = \mathbf{T} \cdot \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3) \cdot \mathbf{T}^{-1}$.

3.3.3 Data enforced tangent

Recalling the definition of the tangent

$$\boldsymbol{C} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{I} - \gamma \boldsymbol{N} \otimes \boldsymbol{N}, \tag{3.47}$$

we found a formula for the normal $\boldsymbol{N}$ in principal stress space by Eq. (3.46) provided we have information about the initial yield surface $\Phi(\theta)$, cf. Eq. (3.33). To address the latter, we assume a combined tensile-torsion test resulting in stress fields of the form

$$\hat{\boldsymbol{\sigma}}_i = \begin{pmatrix} \hat{\sigma}_{11,i} & 0 & 0 \\ 0 & 0 & \hat{\sigma}_{23,i} \\ 0 & \hat{\sigma}_{23,i} & 0 \end{pmatrix}. \tag{3.48}$$

By variation of the ratio $\hat{\sigma}_{23}/\hat{\sigma}_{11}$, it is possible to obtain data points $\{(\rho_i,\theta_i)\}_{i=1}^{n_e}$ lying on the initial yield surface, illustrated in Fig. 3.3.3. By this means, it is possible to approximate the yield function $\Phi(\theta)$ by choosing an appropriate interpolation method, e.g., multilinear polynomial, spline, or nearest-neighbor interpolation.

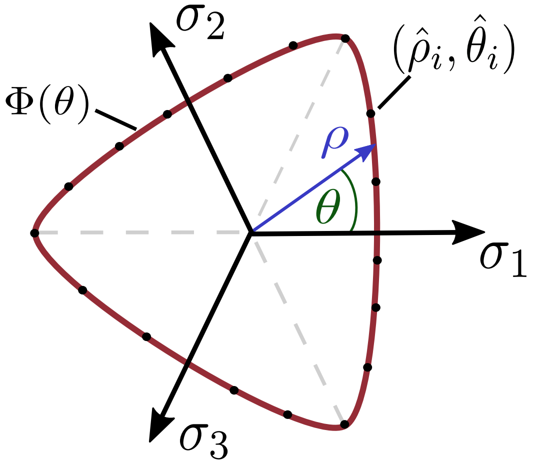


Figure 3.3.3: Schematic illustration of data points $(\hat{\rho}_i,\hat{\theta}_i)$ which are gained by combined tensile-torsion test lying on the initial yield surface $\Phi(\theta)$.

Thus, the remaining task to calculate the complete tangent operator $\boldsymbol{C}$ is to determine parameter $\gamma \in \mathbb{R}_{\geq 0}$ depending on the hardening variable α, i.e., $\gamma \equiv \gamma(\alpha)$. Regarding this, we assume a data set $\{(\hat{\boldsymbol{\varepsilon}}_i,\hat{\boldsymbol{\sigma}}_i)\}_{i=1}^{n_e}$ consisting of strain stress pairs of the form

$$(\hat{\boldsymbol{\varepsilon}}_i,\hat{\boldsymbol{\sigma}}_i) = \left(\begin{pmatrix} \hat{\varepsilon}_{11,i} & 0 & 0 \\ 0 & \hat{\varepsilon}_{22,i} & 0 \\ 0 & 0 & \hat{\varepsilon}_{22,i} \end{pmatrix}, \begin{pmatrix} \hat{\sigma}_{11,i} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right), \tag{3.49}$$

which can be achieved by a simple tensile test. In addition, we calculate strain and stress increments $(\Delta\hat{\boldsymbol{\varepsilon}}_i,\Delta\hat{\boldsymbol{\sigma}}_i) = (\hat{\boldsymbol{\varepsilon}}_i,\hat{\boldsymbol{\sigma}}_i) - (\hat{\boldsymbol{\varepsilon}}_{i-1},\hat{\boldsymbol{\sigma}}_{i-1})$ having the form

$$(\Delta\hat{\boldsymbol{\varepsilon}}_i,\Delta\hat{\boldsymbol{\sigma}}_i) = \left(\begin{pmatrix} \Delta\hat{\varepsilon}_{11,i} & 0 & 0 \\ 0 & \Delta\hat{\varepsilon}_{22,i} & 0 \\ 0 & 0 & \Delta\hat{\varepsilon}_{22,i} \end{pmatrix}, \begin{pmatrix} \Delta\hat{\sigma}_{11,i} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right). \tag{3.50}$$

Substitution of the increments into the derivative relation (3.22) results in

$$\Delta\hat{\boldsymbol{\sigma}}_i = \lambda \mathrm{tr}(\Delta\hat{\boldsymbol{\varepsilon}}_i)\mathbf{I} + 2\mu\Delta\hat{\boldsymbol{\varepsilon}}_i - \gamma_i \boldsymbol{N}(\theta_i) \otimes \boldsymbol{N}(\theta_i), \tag{3.51}$$

representing an implicit equation for parameter γ_i for $i = 1,\dots,n_e$. As a consequence, calculating the normals $\boldsymbol{N}(\theta_i)$ by Eq. (3.24) and parameters γ_i by Eq. (3.51), we can determine the tangents $\boldsymbol{C}_i$ for the strain stress data in (3.49). In the following, we will exploit how to use this to calculate the tangent for a general given modeling point arising in the context of the data-driven approach.

3.3.4 Adapted data-driven projection $P_{\mathcal{D}}$

As mentioned in Section 3.2.1, the issue of accessibility of data and its corresponding tangent space is crucial. In Section 3.3.2 and 3.3.3, we introduced a way to determine the tangent $\boldsymbol{C}$ for strain stress data obtained by a simple tensile test (3.49) using the Haigh-Westergaard space. To exploit this approach for the data-driven method, we recall the fixed-point mapping $P_{\mathcal{D}}P_{\mathcal{C}}$ in Eq. (3.12), where $P_{\mathcal{C}}$ projects a data state $(\hat{\boldsymbol{z}}, \boldsymbol{C}) \in \mathcal{D}^{\text{ext}}$ to the closest point in the constraint set $\boldsymbol{z} \in \mathcal{C}$ and computes the corresponding strain and stress values by Eq. (3.14) and Eq. (3.15). Additionally, projection $P_{\mathcal{D}}$ finds the closest state $(\hat{\boldsymbol{z}}, \boldsymbol{C})$ in data set $\mathcal{D}^{\text{ext}}$ to the calculated material state $\boldsymbol{z}$. While $P_{\mathcal{C}}$ only consists of solving linear equation system (3.13) and therefore cannot be modified in general, projection $P_{\mathcal{D}}$ is used to adapt the nearest neighbor search in principal stress space.

We start by assigning the data set (3.6) by

$$\mathcal{D} = \bigtimes_{e=1}^{m} \mathcal{D}_e \quad \text{with} \quad \mathcal{D}_e := \{(\hat{\varepsilon}_i, \hat{\boldsymbol{\sigma}}_i)\}_{i=1}^{n_e}, \tag{3.52}$$

with strain and stress points obtained by the tensile test described in (3.49). Based on this, we calculate the invariants

$$J_{2,i} = \frac{\hat{\sigma}_{11,i}^2}{3}, \tag{3.53}$$

$$J_{3,i} = \frac{2\hat{\sigma}_{11,i}^3}{27}, \tag{3.54}$$

providing the corresponding Haigh-Westergaard coordinates (3.29)-(3.31) by

$$\hat{\rho}_i = \sqrt{\frac{2}{3}}\|\hat{\sigma}_{11,i}\|, \tag{3.55}$$

$$\cos\left(3\hat{\theta}_i\right) = \frac{3\sqrt{3}}{2} \cdot \frac{2\hat{\sigma}_{11,i}^3}{27} \cdot 3^{3/2}\hat{\sigma}_{11,i}^{-3} = 1. \tag{3.56}$$

Using that Equation (3.56) yields $\hat{\theta}_i = 0$ and substituting this into relation (3.33), we have an equation for the hardening variable given by

$$\hat{\alpha}_i = \frac{\hat{\rho}_i}{\Phi(0)} = \sqrt{\frac{2}{3}}\frac{\|\hat{\sigma}_{11,i}\|}{\Phi(0)}, \tag{3.57}$$

for $i = 1, \ldots, n_e$. Accordingly, we redefine the extended data set (3.10) by

$$\mathcal{D}^{\text{ext}} = \bigtimes_{e=1}^{m} \mathcal{D}_e^{\text{ext}} \quad \text{with} \quad \mathcal{D}_e^{\text{ext}} := \{(\Delta\hat{\varepsilon}_i, \Delta\hat{\boldsymbol{\sigma}}_i), \hat{\alpha}_i\}_{i=1}^{n_e}, \tag{3.58}$$

with strain and stress increments $(\Delta\hat{\varepsilon}_i, \Delta\hat{\boldsymbol{\sigma}}_i) = (\hat{\varepsilon}_i, \hat{\boldsymbol{\sigma}}_i) - (\hat{\varepsilon}_{i-1}, \hat{\boldsymbol{\sigma}}_{i-1})$. The hardening variable α_i can replace the comparison stress in the transition rules of the data-driven mapping $P_{\mathcal{D}}$. Following the procedure of [101], the data sets $\mathcal{D}_e^{\text{ext}}$ are classified into subsets corresponding to elastic and inelastic material behavior according to Eq. (3.16). In the context of the kinetics of elasto-plasticity, the comparison stress is defined as

$$\sigma_{\text{com}}(\boldsymbol{\sigma}) = \frac{\rho}{\Phi(\theta)} = \alpha, \tag{3.59}$$

where the Haigh-Westergaard coordinates ρ, θ depend on $\boldsymbol{\sigma}$ and $\Phi(\theta)$ is a given approximation of the yield surface, cf. Section 3.3.3. Thus, the yield condition reads

$$\sigma_{\text{com}}(\boldsymbol{\sigma}) \leq \alpha_y, \tag{3.60}$$

with $\alpha_y \in \mathbb{R}_{\geq 1}$ denoting the hardening parameter. Hence, for fixed modeling points $\{\boldsymbol{z}_e\}_{e=1}^m$ the mapping $P_{\mathcal{D}} : \mathcal{C} \to \mathcal{D}^{\text{ext}}$ can be adapted for material state $e = 1, \ldots, m$ by:

1. determine the Haigh-Westergaard coordinates ρ_e, θ_e
2. check yield condition and assign index
$$p = \begin{cases} \text{elastic}, & \text{if } \alpha_e < \alpha_{\text{y},e}, \\ \text{inelastic}, & \text{otherwise}, \end{cases} \qquad \text{with } \sigma_{\text{com}}(\boldsymbol{\sigma}_e) = \alpha_e; \tag{3.61}$$
3. if $p \equiv$ elastic:
assign tangent as elastic stiffness matrix i.e.
$$\boldsymbol{C}_e = \boldsymbol{C}_e^{\text{el}}; \tag{3.62}$$
4. if $p \equiv$ inelastic:
   a) set new yield condition
   $$\alpha_{\text{y},e} \equiv \sigma_{\text{com}}(\boldsymbol{\sigma}_e); \tag{3.63}$$
   b) find closest data point $(\Delta\hat{\boldsymbol{\varepsilon}}_e, \Delta\hat{\boldsymbol{\sigma}}_e, \hat{\alpha}_e)$ by
   $$\underset{(\Delta\hat{\boldsymbol{\varepsilon}}_i, \Delta\hat{\boldsymbol{\sigma}}_i, \hat{\alpha}_i) \in \mathcal{D}_e^{\text{ext},\, p}}{\arg\min} \|\alpha_e - \hat{\alpha}_i\|_2; \tag{3.64}$$
   c) calculate the diagonal matrix $\boldsymbol{\sigma}_e^D$ of $\boldsymbol{\sigma}_e$ containing principal stresses $\sigma_{1,e} \geq \sigma_{2,e} \geq \sigma_{3,e}$ and the corresponding transformation matrix $\mathbf{T}_e$ satisfying $\boldsymbol{\sigma}_e^D = \mathbf{T}_e^{-1}\boldsymbol{\sigma}_e\mathbf{T}_e$
   d) calculate normal vector in octahedral plane i.e.
   $$\hat{\boldsymbol{N}}(\theta_e) = \frac{1}{\sqrt{\rho(\theta_e)^2 + \rho'(\theta_e)^2}} \left[\boldsymbol{s}_e + \sqrt{\frac{2}{3}}\rho'(\theta_e) \begin{pmatrix} \sin(\theta_e) \\ -\cos\left(\frac{\pi}{6} - \theta_e\right) \\ \cos\left(\frac{\pi}{6} + \theta_e\right) \end{pmatrix} \right] \tag{3.65}$$
   and transform it into the Cartesian coordinate system
   $$\boldsymbol{N}_e = \mathbf{T}_e \cdot \text{diag}(\hat{\boldsymbol{N}}) \cdot \mathbf{T}_e^{-1}; \tag{3.66}$$
   e) determine parameter γ_e using the equation
   $$\Delta\hat{\boldsymbol{\sigma}}_e = \lambda_e \text{tr}(\Delta\hat{\boldsymbol{\varepsilon}}_e)\mathbf{I} + 2\mu_e\Delta\hat{\boldsymbol{\varepsilon}}_e - \gamma_e\boldsymbol{N}_e \otimes \boldsymbol{N}_e; \tag{3.67}$$
   f) assign tangent as
   $$\boldsymbol{C}_e = \boldsymbol{C}_e^{\text{el}} + \gamma_e\boldsymbol{N}_e \otimes \boldsymbol{N}_e; \tag{3.68}$$
5. set the closest data point $(\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e)$ to modeling point $\boldsymbol{z}_e$ as $(\boldsymbol{z}_e, \boldsymbol{C}_e)$.

The first two steps map the transition rules to the octahedral plane and the corresponding data sets. Step 3 and 4 assign the tangent stiffness matrix distinguishing between the elastic or inelastic assignment. While Step 3 maps the tangent to the elastic stiffness matrix, Step 4 defines a new yield limit and calculates the normal in the octahedral plane, finding the closest point in the data set and using Eq. (3.46). The inelastic stiffness matrix (3.23) is then obtained by a coordinate transformation and Eq. (3.51). Step 5 utilizes that each material state that satisfies the tangent relation within the principal stress space is encompassed by the yield surface, which has the potential to evolve during the process of hardening. We capitalize on the utilization of such states as data points within the mapping function $P_{\mathcal{D}}$. This enables us to derive a more precise determination of the subsequent material state.
Due to the definition of the adapted projection $P_{\mathcal{D}}$ and the usage of the tangent-space structure in $P_{\mathcal{C}}$, the enhanced paradigm increases efficiency compared to the classical data-driven algorithms [30, 101]. This can be shown assuming a given data state $(\hat{\boldsymbol{z}}^{k+1}, \boldsymbol{C}^{k+1}) = P_{\mathcal{D}}(\boldsymbol{z}^k)$ with material state $\boldsymbol{z}^k = P_{\mathcal{C}}(\hat{\boldsymbol{z}}^k, \boldsymbol{C}^k)$ obtained at the k-th fixed-point iteration. Using compatibility condition (3.1) and equilibrium condition (3.2), it follows that the linear equation system (3.13) of projection $P_{\mathcal{C}}(\hat{\boldsymbol{z}}^{k+1}, \boldsymbol{C}^{k+1})$ at iteration $k+1$ reads

$$\begin{aligned}
\left(\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{C}_e^{k+1} \boldsymbol{B}_e\right) \boldsymbol{u}^{k+1} &= \boldsymbol{f} - \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T (\hat{\boldsymbol{\sigma}}_e^{k+1} - \boldsymbol{C}_e^{k+1} \hat{\boldsymbol{\varepsilon}}_e^{k+1}) && (3.69)\\
&= \boldsymbol{f} - \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T (\boldsymbol{\sigma}_e^{k} - \boldsymbol{C}_e^{k+1} \boldsymbol{\varepsilon}_e^{k}) && (3.70)\\
&= \boldsymbol{f} - \boldsymbol{f} + \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{C}_e^{k+1} \boldsymbol{\varepsilon}_e^{k} && (3.71)\\
&= \left(\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{C}_e^{k+1} \boldsymbol{B}_e\right) \boldsymbol{u}^{k}, && (3.72)
\end{aligned}$$

which yields $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k$ and therefore $\boldsymbol{z}^{k+1} = \boldsymbol{z}^k$. Consequently, only one fixed-point iteration in Eq. (3.12) is required. The detailed adapted data-driven scheme $P_{\mathcal{D}} P_{\mathcal{C}}$ is summarized in Algorithm 3.

The methodology efficiently deals with inelasticity while considering isotropic hardening effects by formulating the problem within the Haigh-Westergaard space. The essential contribution of this approach lies in its utilization of coordinates in the principal strain and stress space to define the yield function and the corresponding normal. This choice offers notable advantages, such as reducing the amount of data required for storage and manipulation and minimizing the computational effort involved in the projection steps compared to conventional data-driven approaches. However, the proposed approach is mainly designed to accommodate symmetric yield surfaces rather than explicitly addressing anisotropic hardening effects. In cases of materials with anisotropic hardening behavior, additional adaptations or extensions may be necessary to accurately capture the complex hardening effects within the framework of the approach. The next section demonstrates the performance of the proposed scheme via a numerical example employing elasto-plasticity with isotropic hardening.

3.4 Numerical result for a $3D$ benchmark

This section illustrates the performance of the adapted data-driven solver extended by the tangential space information in Haigh-Westergaard space in a typical benchmark, considering stress analysis of elasto-plastic material with non-linear isotropic hardening based on experimental measurements synthetically simulated. In our scope, synthetic data consists of strain-stress points created numerically using a material model rather than obtained by actual experimental measurements. We restrict the simulation to noise-free synthetic data sets. However, experimental data isÂ generally noisy and includes outliers. This issue can be treated using noise reduction algorithms such as tensor voting [105], Kalman filtering [75], and deep learning-based methods.

The application and the numerical performance will be demonstrated by investigating the impact of increasing the number of tensile data on the convergence of the proposed data-driven scheme. The significance of the quantity of tension-torsion data points is secondary because the accuracy of the approximation of the yield surface is significantly dependent on the method utilized, such as polynomial or spline interpolation, nearest neighbor approaches, or machine learning methods.
The root-mean-square error between the data-driven strain and stress solution $\boldsymbol{z}^k$ and its corresponding reference solution $\boldsymbol{z}^{k,\text{ref}}$ will be calculated using

$$\text{RMSD}(\boldsymbol{z})^2 = \frac{\sum_{k=0}^{T} \text{Error}(\boldsymbol{z}^k)^2}{T}, \tag{3.73}$$

where $T \in \mathbb{N}$ is the number of total loading steps, $\boldsymbol{z}_e^k = (\boldsymbol{\varepsilon}_e^k, \boldsymbol{\sigma}_e^k)$ the local data-driven states and $\boldsymbol{z}_e^{k,\text{ref}} = (\boldsymbol{\varepsilon}_e^{k,ref}, \boldsymbol{\sigma}_e^{k,\text{ref}})$ the local reference states at step $k \leq T$. The error is given by

$$\text{Error}(\boldsymbol{z}^k)^2 = \frac{\sum_{e=1}^{m} w_e \|\boldsymbol{z}_e^k - \boldsymbol{z}_e^{k,\text{ref}}\|_e^2}{\sum_{e=1}^{m} w_e \|\boldsymbol{z}_e^{k,\text{ref}}\|_e^2}, \tag{3.74}$$

with $\| \cdot \|_e$ given by the definition in Eq. (3.9). The reference states are computed using an iterative return mapping algorithm embedded in a Newton-Raphson global loop restoring equilibrium.

3.4.1 Plate with a circular hole

In this benchmark, we investigate a $3D$ plate with a circular hole subjected to an increasing extension $\bar{u}$ in the length direction and a uniformly distributed load p over the thickness direction. Due to the symmetry of geometry and load, only one-quarter of the system is modeled. The geometry, boundary conditions, and loading are chosen according to a similar test presented in [106] and illustrated in Fig. 3.4.1. The side lengths of the strip are equal to $a = b = 5\,\text{m}$, and the thickness is $c = 2\,\text{m}$. The radius of the hole is $r = 5\,\text{m}$. Displacements are fixed at the quarter plates' left surface $x = 0$ in x-direction and at the bottom surface $y = 0$ in y-direction. For $z = 0$, the displacements are fixed in z-direction. The corresponding boundary conditions read as follows:

$$\begin{cases} u_x = 0, & \text{if } x = 0; \\ u_y = 0, & \text{if } y = 0; \\ u_z = 0 & \text{if } z = 0; \\ (u_x, u_y, u_z) = (\bar{u}_x, \bar{u}_y, \bar{u}_z) & \text{if } x = a, \end{cases} \tag{3.75}$$

where u_x, u_y, u_z and $\bar{u}_x, \bar{u}_y, \bar{u}_z$ are the displacements in x, y and z-directions, respectively.

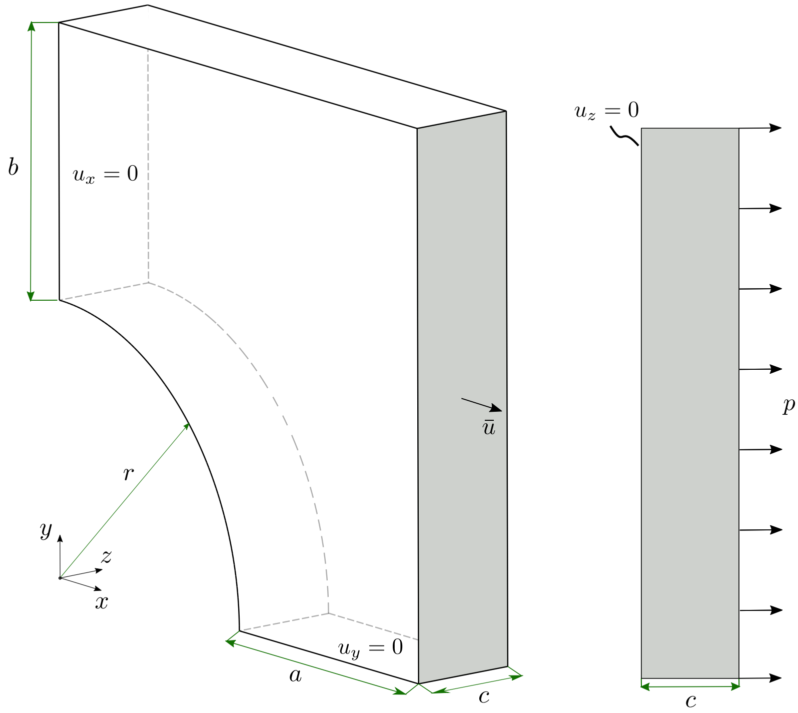


Figure 3.4.1: Boundary conditions and dimensions of a quadratic plate under increasing extension $\bar{u}$ on the grey area and uniformly distributed pressure p over the thickness direction.

Material parameters:

This setting considers an elasto-plastic material with non-linear isotropic hardening. The applied material parameters are Young's modulus E, Poisson's ratio ν, and elasticity tensor given by

$$\boldsymbol{C}^{\text{el}} = \left(\kappa - \frac{2}{3}G\right) \mathbf{I} \otimes \mathbf{I} + 2G\,\mathbb{I}, \tag{3.76}$$

with bulk and shear moduli $\kappa = \frac{E}{3(1-2\nu)}$ and $G = \frac{E}{2(1+\nu)}$. A power law describes the non-linear isotropic hardening by

$$\sigma_{\text{y}}(\bar{\varepsilon}^p) = \left(1 - \frac{1}{3}\tan 30^\circ\right)\left(\sigma_0 + H(\bar{\varepsilon}^p)^{1/h}\right). \tag{3.77}$$

where $\sigma_0 \in \mathbb{R}_+$ is the initial yield limit, $H \in \mathbb{R}_+$ is the hardening modulus, $h \in (0,1)$ is the hardening exponent and $\bar{\varepsilon}^p \in \mathbb{R}_+$ defines equivalent plastic strain. For the simulation, we investigate the yield criterion

$$F(\boldsymbol{\sigma}) \leq \sigma_{\text{y}}(\bar{\varepsilon}^p), \tag{3.78}$$

with

$$F(\boldsymbol{\sigma}) := \frac{\rho\sqrt{3}}{2\sqrt{2}}\left(1 + \frac{1}{k} - \left(1 - \frac{1}{k}\right)\cos(3\theta)\right), \tag{3.79}$$

where ρ, θ are Haigh-Westergaard Coordinates depending on $\boldsymbol{\sigma}$ and $k \in \mathbb{R}_+$ is a ratio that controls the dependence of the yield surface on the principal stress. One can notice that for $k = 1$ Equation (3.78) represents a von Mises yield surface, see Fig. 3.4.2, case (a). In addition, the initial yield function can be written as

$$\Phi(\theta) = \sigma_{\mathrm{y}}(0) \left(\frac{\sqrt{3}}{2\sqrt{2}} \left[1 + \frac{1}{k} - \left(1 - \frac{1}{k} \right) \cos(3\theta) \right] \right)^{-1} . \tag{3.80}$$

Synthetic data:

The synthetic data in this benchmark setting consists of a combined tension-torsion test, Eq. (3.48), and a single tension test, Eq. (3.49). The points $\hat{\boldsymbol{\sigma}}_i$ of the tension-torsion test depend only on the two components $\hat{\sigma}_{11,i}$ and $\hat{\sigma}_{23,i}$ with $i = 1, \ldots, n_{1,e}$. Since the data is used to approximate the initial yield surface function $\Phi(\theta)$, it is enough to simulate points lying on it. For this, we define uniformly distributed random components $\hat{\sigma}_{11,i} \in [-\sigma_{\mathrm{y}}(0), k\sigma_{\mathrm{y}}(0)]$. The points $\hat{\boldsymbol{\sigma}}_i$ lying on the yield surface are then determined by $\hat{\sigma}_{11,i}$ and $\hat{\sigma}_{23,i}$ obtained by solving

$$\frac{\sqrt{\hat{\sigma}_{11,i}^2 + 3\hat{\sigma}_{23,i}^2}}{2k} \left(1 + k + (1-k) \frac{\left(\hat{\sigma}_{11,i}^3 - 9\hat{\sigma}_{11,i}\, \hat{\sigma}_{23,i}^2 \right)}{\left(\hat{\sigma}_{11,i}^2 + 3\hat{\sigma}_{23,i}^2 \right)^{\frac{3}{2}}} \right) = \sigma_{\mathrm{y}}(0). \tag{3.81}$$

The second data set is generated by simulating a uniaxial tensile test subject to predefined loading paths. The resulting data points $(\hat{\varepsilon}_i, \hat{\boldsymbol{\sigma}}_i)$ are then used to establish the data set (3.58) consisting of points $(\Delta\hat{\varepsilon}_i, \Delta\hat{\boldsymbol{\sigma}}_i, \hat{\alpha}_i)$ with $i = 1, \ldots, n_{2,e}$. The initial hardening parameter of the data-driven comparison stress in Eq. (3.59) is given by $\alpha_{y,e} = 1$ for all material states $e = 1, \ldots, m$.

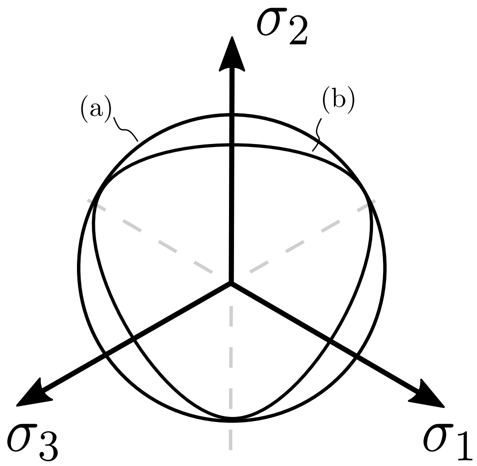


Figure 3.4.2: Visualization of yield surface function $\Phi(\theta)$ for (a) $k = 1$ and (b) $k = 0.75$.

Convergence and results:

The load p increases from 0.0 to $3.0 \cdot 10^7$ Pa, decreases to 0.0 Pa and then increases to $3.5 \cdot 10^7$ Pa using 150 time steps per path with a constant step size of $\Delta t = 1$. Correspondingly, the applied total displacement $\bar{\boldsymbol{u}}$ on the lateral side increases to $0.3\,\mathrm{m}$, decreases to $0.0\,\mathrm{m}$ and then increases to $0.4\,\mathrm{m}$. The system is discretized by $10-$nodal tetrahedron elements (P2) with quadratic ansatz functions on a mesh of element size 1460. The exact material parameters used for the reference solution and synthetic data are given in Table 3.1. For $k = 0.75$, the yield surface in principle stress plane is depicted in Fig. 3.4.2, case (b). Regarding the data-driven simulation, we fix the number of tensile-torsion data by $n_{1,e} = 50$ and use simple spline interpolation to approximate the yield surface $\Phi(\theta)$. The displacement and maximum principle stress convergence are then investigated for a small tensile data set of size $n_{2,e} = [10, 20, 40, 60, 80, 90]$ and a more extensive data set of the size of $n_{2,e} = 10^j$ with $j = 2, \dots, 5$. Each set of tensile test data is simulated using a different number of loading paths n_p chosen from the set $[1, 2, 4, 6, 8, 10]$. Therefore, we investigate $12 \cdot 6 = 52$ different data sets. For clarity, $n_{2,e} = 10$ and $n_p = 1$ is the data set defined by a single loading path with 10 data points. The data set consists of two loading paths containing 10 data points represented by $n_{2,e} = 10$ and $n_p = 2$. The last data set consists of 10 loading paths with 10^5 data points i.e., $n_{2,e} = 10^5$ and $n_p = 10$. Figure. 3.4.3 illustrates exemplary the generation structure of the synthetic data sets.

$E\,[\mathrm{Pa}]$	$\nu\,[-]$	$H\,[\mathrm{Pa}]$	$\sigma_0\,[\mathrm{Pa}]$	$h\,[-]$	$k\,[-]$
$3 \cdot 10^{10}$	0.2	$2.5 \cdot 10^9$	$3 \cdot 10^8$	2	0.75

Table 3.1: Material parameters

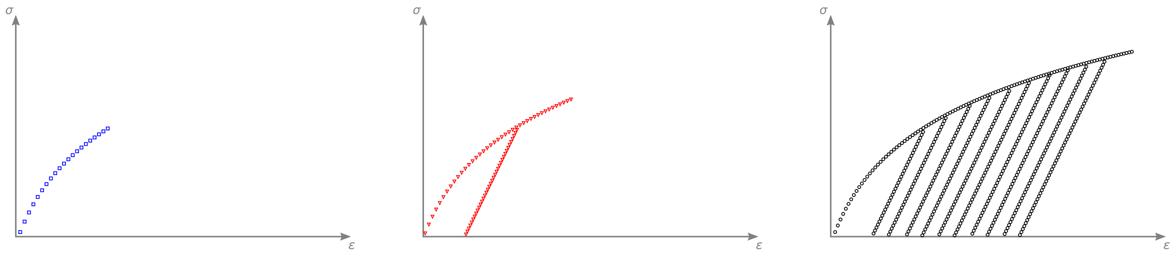


Figure 3.4.3: An illustrative visualization of synthetic tensile data. The first data set (blue, □) is simulated using a single loading path with 10 data points, and the second data set (red, △) consists of two loading paths containing 10 data points. The third data set (black, ○) consists of 10 loading paths of 10^5 data points.

The plot in Fig. 3.4.4 shows the convergence of the error corresponding to the increased data size and the number of loading paths. The bias results from the chosen number of time steps. During the simulation, we noticed that increasing the increment steps decreases the error significantly. Figure 3.4.5 illustrates the occurring displacement in the elasto-plastic plate with a circular hole. In addition, it compares the absolute errors of displacement to the reference solution at internal nodes. Figure 3.4.6 shows the interpolated maximum principle stresses in $[\mathrm{Pa}]$ at each integration point for time step $t = 150, 300$ and 450. Furthermore, we plot the relative error compared to the reference solution. We conclude that increasing the number of data and the number of loading paths decreases the RMSD error, corresponding to the data-driven convergence analysis of [101].

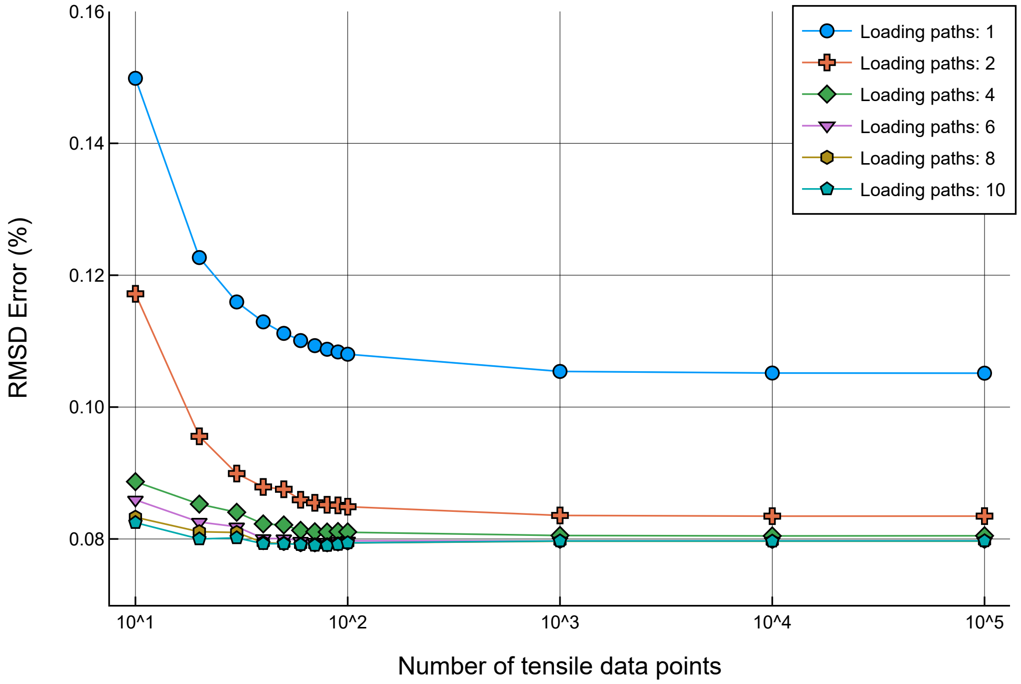
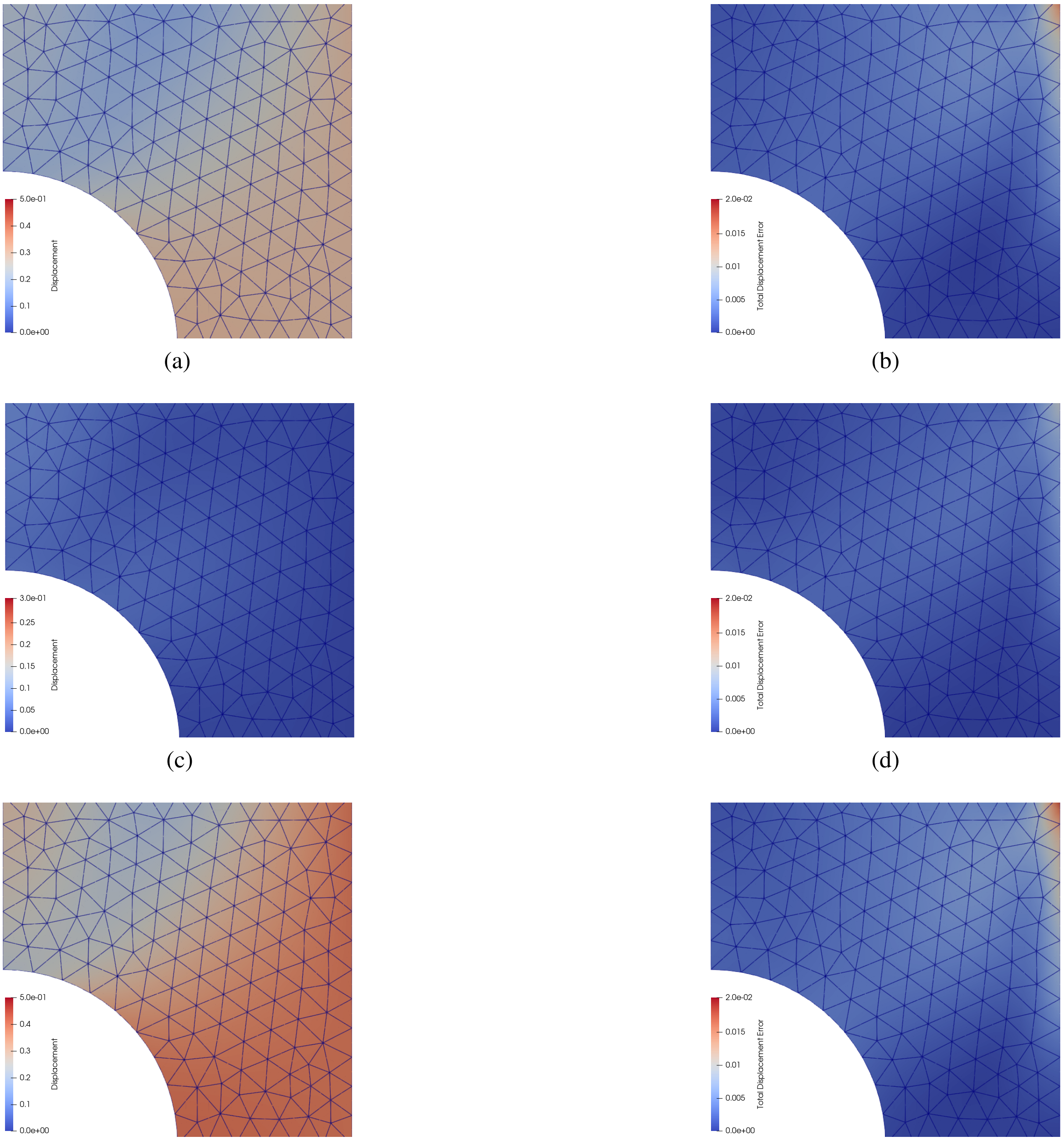


Figure 3.4.4: RMSD error of the boundary value problem using the adapted data-driven transition mapping to the reference solution based on the exact material model. The graphs are illustrated concerning the size of tensile data and its corresponding number of loading paths.

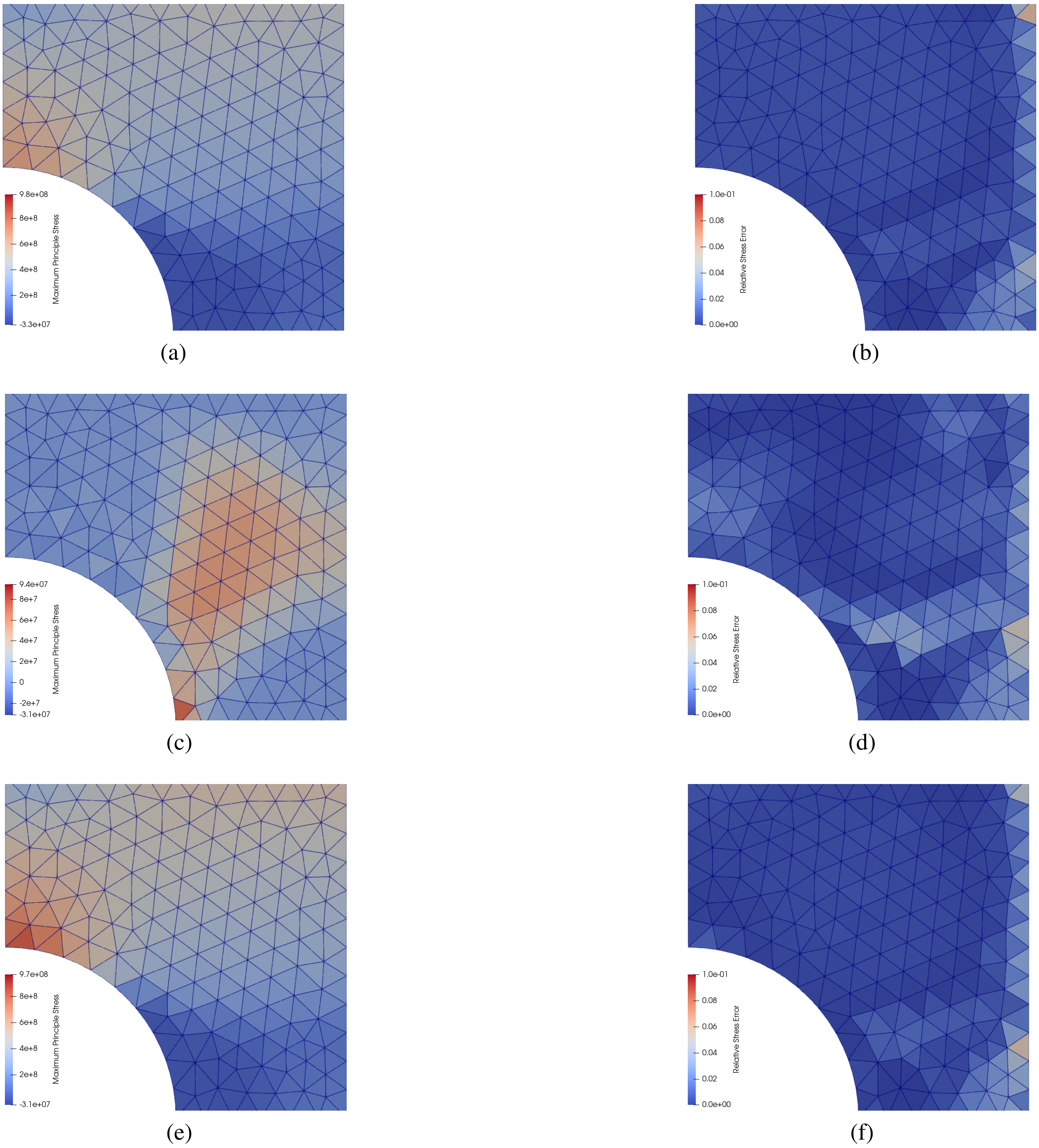


(a) (b)

(c) (d)

(e) (f)

Figure 3.4.5: Contour plot of maximum displacement of the boundary value problem using the adapted data-driven transition mapping. (a, c, e) Maximum displacement and corresponding (b, e, f) absolute errors compared to reference solution at each internal node at time step $t = 150, 300, 450$. The number of loading paths simulating the tensile test is 10 with 10^5 data points.



(a) (b)

(c) (d)

(e) (f)

Figure 3.4.6: Contour plot of maximum principal stress of the boundary value problem using the adapted data-driven transition mapping. (a, c, e) Maximum principal stress and corresponding (b, e, f) relative errors compared to reference solution at each material point in [Pa] at time step $t = 150, 300, 450$. The number of loading paths simulating the tensile test is 10 with 10^5 data points.

3.5 Conclusion

We present using the Haigh-Westergaard space to obtain data points from measurements for the model-free data-driven inelasticity extended by tangent space. Even though the original approach is sufficient for ideal data, the issue of data accessibility and the associated tangent space arises. In particular, data points of inelastic materials could only be acquired through impracticable sample testing in a comprehensive scope of loading directions.

This paper addresses the issue of data accessibility for isotropy guaranteeing the material's loading direction independence. Thus, the tangent space is specified by the material's hardening behavior and the yield surface normal. For the former, we employed data from a straightforward tensile test. We adopted Haigh-Westergaard coordinates for the latter to project the data-driven approach onto the octahedral plane. Then, a combined tension-torsion test provides sufficient information about the underlying material yield surface for approximating the characteristic function. The resulting data-driven method minimizes the distance to the tensile test data and calculates the associated tangent stiffness in the Haigh-Westergaard space, subject to compatibility and equilibrium constraints. The resulting scheme leads to much greater efficiency, especially using material points as data states reduces the fix-point problem to only one iteration.

The application and its numerical performance have been demonstrated on a $3D$ isotropic elasto-plastic benchmark with non-linear hardening. The accuracy improves for larger tensile data sets, and the convergence rate correlates with the convergence analysis of data-driven inelasticity. We neglected the accounting regarding the number of tensile-torsion data points since the quality of the yield surface approximation depends highly on the used method, e.g., polynomial or spline interpolation, nearest neighbor approaches, or machine learning methods. We have limited the simulation to synthetic noise-free data sets. However, experimental data is generally noisy and includes outliers. This issue can be treated using noise reduction algorithms such as tensor voting, Kalman filtering, and deep learning-based methods.

The developments of the data-driven paradigm propose crucial future research areas in machine-learning methods, particularly physics-informed neural networks. By specifying suitable loss functions, these networks can be trained to fulfill training data and discover optimal solutions for given physics-governing equations. Since the data-driven method bypasses the step of material modeling but still relies on solving governing equations, a combined formulation of the model-free data-driven and the physics-informed neural network method is possible.

Appendix 3.A Algorithm of the modified data-driven solver

Algorithm 3 Data-driven solver at time step t using Haigh-Westergaard coordinates

```
Input: matrices $\{\boldsymbol{B}_e\}_{e=1}^m$, weights $\{w_e\}_{e=1}^m$, load $\boldsymbol{f}$, Lamé constants $\lambda_e, \mu_e$
Data: tensile data $\mathcal{D}_e^{\text{ext}}$, $\Phi_e(\theta)$ obtained through tension-torsion data $\{(\hat{\rho}_i, \hat{\theta}_i)\}_{i=1}^{n_e}$
  procedure DATA-DRIVEN SOLVER
      if $t = 1$ then                                                   ▷ Initialize variables
          for all $e = 1, \dots, m$ do
              $\boldsymbol{C}_e^{\text{el}} = \lambda_e \mathbf{I} \otimes \mathbf{I} + 2\mu_e \mathbb{I}$                  ▷ Elastic stiffness
              $\hat{\boldsymbol{z}}_e = ((\hat{\boldsymbol{\varepsilon}}_e, \hat{\boldsymbol{\sigma}}_e), \boldsymbol{C}_e) \leftarrow ((\mathbf{0}, \mathbf{0}), \boldsymbol{C}_e^{\text{el}})$       ▷ Data state
              $\alpha_{y,e} \leftarrow 1$                                ▷ Transition variable
          end for
      end if
                                                                        ▷ Projection $P_{\mathcal{D}} P_{\mathcal{C}}$
      $\{\boldsymbol{z}_e\}_{e=1}^m = P_{\mathcal{C}}(\{\hat{\boldsymbol{z}}_e\}_{e=1}^m)$
      $\{\hat{\boldsymbol{z}}_e, \boldsymbol{C}_e\}_{e=1}^m = P_{\mathcal{D}}(\{\boldsymbol{z}_e\}_{e=1}^m)$
      $t \leftarrow t + 1$
  end procedure
```

Projection $P_{\mathcal{C}}(\hat{\boldsymbol{z}})$

```
Solve equation system:
    $$\left(\sum_{e=1}^{m} w_e \boldsymbol{B}_e^T \boldsymbol{C}_e \boldsymbol{B}_e\right) \boldsymbol{u} = \boldsymbol{f} - \sum_{e=1}^{m} w_e \boldsymbol{B}_e^T (\hat{\boldsymbol{\sigma}}_e - \boldsymbol{C}_e \hat{\boldsymbol{\varepsilon}}_e)$$
for all $e = 1, \dots, m$ do
    $\boldsymbol{\varepsilon}_e = \boldsymbol{B}_e \boldsymbol{u}$,
    $\boldsymbol{\sigma}_e = \hat{\boldsymbol{\sigma}}_e + \boldsymbol{C}_e(\boldsymbol{\varepsilon}_e - \hat{\boldsymbol{\varepsilon}}_e)$
end for
return $\{\boldsymbol{z}_e\}_{e=1}^m = \{(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e)\}_{e=1}^m$
```

Projection $P_{\mathcal{D}}(\boldsymbol{z})$

```
for all $e = 1, \dots, m$ do
    $\boldsymbol{\sigma}_e^D, \mathbf{T}_e = \textsc{PrincipalStress}(\boldsymbol{\sigma}_e)$    ▷ Diagonalization
    $(\rho_e, \theta_e) \leftarrow \left(2\sqrt{J_{2,e}}, \frac{1}{3}\arccos\left(\frac{3\sqrt{2}}{2} J_{3,e} J_{2,e}^{-3/2}\right)\right)$    ▷ Haigh-Westergaard
    $\alpha_e = \frac{\rho_e}{\Phi_e(\theta_e)}$
    if $\alpha_e \leq \alpha_{\mathrm{y},e}$ then    ▷ Transition rule
        $\boldsymbol{C}_e \leftarrow \boldsymbol{C}_e^{\mathrm{el}}$
    else
        $(\Delta\hat{\boldsymbol{\varepsilon}}_e, \Delta\hat{\boldsymbol{\sigma}}_e, \hat{\alpha}_e) = \underset{(\Delta\hat{\boldsymbol{\varepsilon}}_i, \Delta\hat{\boldsymbol{\sigma}}_i, \hat{\alpha}_i) \in \hat{\mathcal{D}}_e^{\mathrm{ext}}}{\arg\min} \|\alpha_e - \hat{\alpha}_i\|_2$
        $\rho_e' = \alpha_e \Phi_e'(\theta_e)$
        $\boldsymbol{N}_e, \gamma_e = \textsc{Normal}((\rho_e, \rho_e', \theta_e), (\Delta\hat{\boldsymbol{\varepsilon}}_e, \Delta\hat{\boldsymbol{\sigma}}_e, \hat{\alpha}_e), \mathbf{T}_e, \lambda_e, \mu_e)$
        $\boldsymbol{C}_e \leftarrow \boldsymbol{C}_e^{\mathrm{el}} - \gamma_e \boldsymbol{N}_e \otimes \boldsymbol{N}_e$
        $\alpha_{\mathrm{y},e} \leftarrow \alpha_e$
    end if
end for
return $\{(\boldsymbol{z}_e, \boldsymbol{C}_e)\}_{e=1}^m$
```

Functions

```
function $\textsc{PrincipalStress}(\boldsymbol{\sigma})$
    return diagonal matrix $\boldsymbol{\sigma}^D$ containing principal stresses $\sigma_1 \geq \sigma_2 \geq \sigma_3$ and corresponding transformation matrix $\mathbf{T}$ satisfying $\boldsymbol{\sigma}^D = \mathbf{T}^{-1}\boldsymbol{\sigma}\mathbf{T}$
end function
function $\textsc{Normal}((\rho, \rho', \theta), (\Delta\hat{\varepsilon}, \Delta\hat{\boldsymbol{\sigma}}, \hat{\alpha}), \mathbf{T}, \lambda, \mu)$
    $\hat{\boldsymbol{N}} = \frac{\sqrt{2}}{\sqrt{3}\sqrt{\rho^2 + \rho'^2}} \left[\rho \begin{pmatrix} \cos(\theta) \\ \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) \end{pmatrix} + \rho' \begin{pmatrix} \sin(\theta) \\ -\cos\left(\frac{\pi}{6} - \theta\right) \\ \cos\left(\frac{\pi}{6} + \theta\right) \end{pmatrix}\right]$
    $\boldsymbol{N} \leftarrow \mathbf{T} \cdot \mathrm{diag}(\hat{\boldsymbol{N}}) \cdot \mathbf{T}^{-1}$
    $\gamma \leftarrow \Delta\hat{\boldsymbol{\sigma}} = \lambda \mathrm{tr}(\Delta\hat{\varepsilon})\mathbf{I} + 2\mu\Delta\hat{\varepsilon} - \gamma \boldsymbol{N} \otimes \boldsymbol{N}$,
    return $\boldsymbol{N}, \gamma$
end function
```

4 Article 3: A physics-informed GAN framework based on model-free data-driven computational mechanics

This article was published as:

CIFTCI, K., & HACKL, K. (2024). A physics-informed GAN framework based on model-free data-driven computational mechanics. Computer Methods in Applied Mechanics and Engineering, 424, 116907.

Disclosure of the individual authors' contributions to the article:

The manuscript results from the research between K. Ciftci and K. Hackl in model-free data-driven computational mechanics. Inspired by the foundational concepts of physis-informed neural networks, K. Ciftci incorporated the data-driven paradigm with physics-informed generative adversarial networks (GANs) to address the issue of physics-informed neural networks (PINNs) having performance problems. Since GANs are based on data, he believed the model-free data-driven approach, introduced by Kirchdoerfer and Ortiz, would be a suitable approach. K. Hackl provided invaluable expertise, guiding K. Ciftci in refining and detailing the idea. Together, they created a method where the generator uses physical rules, and the discriminator checks the output using strain-stress data. K. Ciftci wrote the paper and conducted all the numerical simulations. The final manuscript has received the endorsement of both authors for publication.

Abstract: Model-free data-driven computational mechanics, first proposed by Kirchdoerfer and Ortiz, replace phenomenological models with numerical simulations based on sample data sets in strain-stress space. In this study, we integrate this paradigm within physics-informed generative adversarial networks (GANs). We enhance the conventional physics-informed neural network framework by implementing the principles of data-driven computational mechanics into GANs. Specifically, the generator is informed by physical constraints, while the discriminator utilizes the closest strain-stress data to discern the authenticity of the generator's output. This combined approach presents a new formalism to harness data-driven mechanics and deep learning to simulate and predict mechanical behaviors.

4.1 Introduction

The simulation of boundary value problems typically contains two equations: conservation and constitutive laws. While conservation laws are derived from universal principles, constitutive laws are usually obtained by fitting model parameters to given strain-stress data [98]. Nevertheless, material modeling can be ill-posed and adds uncertainties to the solutions, particularly in highly complex systems. The model-free data-driven method, introduced by Kirchdoerfer and Ortiz [30], bypasses the step of material modeling, incorporating experimental data directly into the numerical simulations of boundary-value problems. The data-driven scheme computes the closest point in the material data set consistent with the problem's compatibility and equilibrium condition. Consequently, it provides an alternative formulation of the classical initial-boundary-value problem based on nearest-neighbor clustering.

The approach has been fine-tuned for diverse applications: from non-linear elasticity [30, 33, 29, 38, 96] to dynamics [42] and finite strain [37]. It's also been adapted for material data identification [65], non-local mechanics [56], electro-mechanical problems [55], homogenization schemes [61], and model-driven coupling [54]. Recent studies by Poelstra et al. [53] have expanded this framework to address evolutionary problems in solid mechanics. Similarly, Bartel et al. [52] have developed a data-driven approach for plasticity using history surrogates in truss structures. Ibañez et al. [44, 45] refined the approach using a manifold learning method that maps data into a lower-dimensional space to use the locally linear embeddings. Eggersmann et al. [48] presented a second-order data-driven approach that uses tensor voting [97] to obtain point-wise tangent space, enabling the search for additional states close to the original data. For inelastic boundary value problems, Eggersmann et al. [49] include local histories in the data set to investigate materials with memory. Karapiperis et al. [58] have also suggested a variation of the scheme, considering multiscale modeling. In addition, we recently developed a paradigm incorporating the tangent space into the distance-minimizing data-driven formulation and classifies the underlying data structure into subsets according to various material behaviors [101]. The framework features a parametrization of the material history and an optimal sampling of the mechanical system's state space. In line with these developments, [107] emphasizes the essential role of the tangent space within this computational framework. The authors demonstrate the optimal convergence rate for fundamental scenarios and introduce strategies to lower computational costs and enhance solutions' accuracy.

The paradigm's dependence on the nearest-neighbor clustering of data points proposes research areas in machine-learning methods, particularly Artificial Neural Networks (ANNs), that are known to approximate any continuous function for appropriate network parameters [108, 109]. The flexibility and quality of neural networks led to success in a wide range of applications, e.g., image recognition [110], language processing [111], or generative modeling [21, 112]. An extension to neural networks is physics-informed deep learning, successfully used in solving physical-related problems such as fluid mechanics [113, 114], aerodynamics [115, 116], shell structures [117] or material science [118, 119]. Physics-Informed Neural Networks (PINNs) can be trained to fulfill training data and learn optimal solutions for allocated physics-governing equations by specifying appropriate loss functions [120, 78]. The physics-based loss competes against a data-based loss, which is needed to provide fundamental knowledge of the system. Thus, partial differential equations act as additional constraints during network training, resulting in a multi-objective optimization problem. Optimizing data and physics give physics-informed neural networks flexibility in solving forward and inverse problems [116, 114, 115, 121, 122, 123]. The trade-off between the individual losses can be influenced using hyper-parameter [124, 125, 126]. For example, adaptive activation functions [127, 128], or manually weighted losses [129], can improve the quality of the neural network for specific problems. Since PINNs are based on global ansatz functions, the optimization faces challenges in accurately displaying localized effects through training. Henkes et al. [130] present a formidable challenge for methods that rely on a global strategy to investigate adaptive training strategies and domain decomposition, focusing on improving convergence in complex scenarios.

This investigation combines the model-free data-driven approach with Generative Adversarial networks (GANs). In machine learning, GANs have emerged as a powerful tool consisting of two neural networks â€“ the generator, which creates data, and the discriminator, which evaluates the authenticity of the generated data. Through their adversarial game, GANs are adept at generating high-fidelity data, often indistinguishable from actual data [112]. An extension is the integration of physics-informed neural networks with the GAN structure. For instance, the pursuit of robust uncertainty quantification within the framework of PINNs has led to recent methodologies. The PIG-GAN framework [131] harnesses the capabilities of a physics-informed generator to address adversarial uncertainty. On the other hand, the PID-GAN approach [132] uses a physics-informed discriminator, carving out a distinct avenue to achieve reliable uncertainty quantification while maintaining fidelity to the governing physics. Another stride in this direction is the DeqGAN, which offers a unique perspective on PINNs by learning the loss function via generative adversarial networks. This methodology provides a robust avenue for solving the challenges traditionally associated with defining appropriate loss functions for PINNs [133]. In our approach, the generator is a physics-informed neural network, and the discriminator employs the closest strain-stress data to evaluate the authenticity of the generator's results. This synergized methodology matches model-free data-driven computational mechanics and deep learning principles to simulate and predict intricate mechanical behaviors more accurately.

Section 4.2 provides a general setting by introducing the definitions and derivation of the distance-minimizing data-driven computing method based on [29]. Section 4.3 introduces the framework of artificial neural networks and generative adversarial networks. In addition, we propose using a physics-informed GAN to solve the distance-minimizing data-driven problem. Section 4.4 exhibits the performance of the proposed method using a numerical example involving a non-linear elastic in-plane boundary value problem. Finally, Section 4.5 summarizes the results and suggests future research subjects.

4.2 Model-free Data-driven setting

The following will summarize the classical data-driven computational mechanics method for the reader's convenience based on the definitions and formulations in [29]. For a comprehensive exploration of the variational framework, the reader is referred to [134].
We consider an elastic body $\Omega \subset \mathbb{R}^d$ whose internal states are defined by displacement field $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and the compatibility and equilibrium conditions

$$\begin{aligned} \boldsymbol{\varepsilon}(\mathrm{x}) - \nabla^{\mathrm{sym}}\boldsymbol{u}(\mathrm{x}) &= \mathbf{0}, \quad \text{in } \Omega, \\ \nabla \cdot \boldsymbol{\sigma}(\mathrm{x}) - \boldsymbol{f}(\mathrm{x}) &= \mathbf{0}, \quad \text{in } \Omega, \end{aligned} \tag{4.1}$$

and boundary conditions

$$\begin{aligned} \boldsymbol{u}(\mathrm{x}) &= \boldsymbol{g}(\mathrm{x}), && \text{on } \Gamma_D, \\ \boldsymbol{\sigma}(\mathrm{x}) \cdot \boldsymbol{n}(\mathrm{x}) &= \boldsymbol{t}(\mathrm{x}), && \text{on } \Gamma_N, \end{aligned} \tag{4.2}$$

where $\boldsymbol{\varepsilon} : \Omega \to \mathbb{R}^{d\times d}_{\mathrm{sym}}$ is the strain field and $\boldsymbol{\sigma} : \Omega \to \mathbb{R}^{d\times d}_{\mathrm{sym}}$ is the stress field. The boundary Γ of the domain Ω is defined by the Dirichlet (Γ_D) and Neumann (Γ_N) with $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. In addition, $\boldsymbol{f} : \Omega \to \mathbb{R}^d$ is the body force, and $\boldsymbol{g}, \boldsymbol{t}, \boldsymbol{n} : \Gamma \to \mathbb{R}^d$ define the boundary displacement, applied traction and outer normal, respectively.
We define $Z_{\mathrm{loc}} \subset \mathbb{R}^{d\times d}_{\mathrm{sym}} \times \mathbb{R}^{d\times d}_{\mathrm{sym}}$ as the local phase space consisting of pairs $\boldsymbol{z}(\mathrm{x}) = (\boldsymbol{\varepsilon}(\mathrm{x}), \boldsymbol{\sigma}(\mathrm{x}))$ describing the local state of the system at material point x. The global phase space Z is defined as the collection of the state functions, i.e.

$$Z = \{\boldsymbol{z} \,:\, \boldsymbol{z} \in Z_{\mathrm{loc}}\}. \tag{4.3}$$

The data-driven distance-minimization problem, introduced by [30], reads

$$\underset{\boldsymbol{z}\in\mathcal{C}, \hat{\boldsymbol{z}}\in\mathcal{D}}{\arg\min}\, d(\boldsymbol{z}, \hat{\boldsymbol{z}}), \tag{4.4}$$

where $\mathcal{C} \subset Z$ denotes the constraint set defined by

$$\mathcal{C} := \Big\{ \boldsymbol{z} \in Z : \text{(4.1) and (4.2)} \Big\}; \tag{4.5}$$

containing all states fulfilling compatibility and equilibrium. The set $\mathcal{D} \subset Z$ consists of finite data points either obtained by experimental measurements or achieved from elaborate simulations; and is defined by

$$\mathcal{D} = \{\boldsymbol{z} \in Z \,:\, \boldsymbol{z}(\mathrm{x}) \in \mathcal{D}_{\mathrm{loc}}\} \quad \text{with} \quad \mathcal{D}_{\mathrm{loc}} = \{(\boldsymbol{\varepsilon}_i, \boldsymbol{\sigma}_i)\}_{i=1}^{n_e}, \tag{4.6}$$

where $n_e \in \mathbb{N}$ is the number of local data points associated with the material point. The distance $d : Z \times Z \to \mathbb{R}$ is defined by

$$d(\boldsymbol{z}, \hat{\boldsymbol{z}}) := \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|, \tag{4.7}$$

metricized by the norm

$$\|\boldsymbol{z}\|^2 := \int_{\Omega} \left(\frac{1}{2}\boldsymbol{C}\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} + \frac{1}{2}\boldsymbol{C}^{-1}\boldsymbol{\sigma} : \boldsymbol{\sigma} \right) \mathrm{d}\mathrm{x}, \tag{4.8}$$

where $\boldsymbol{C} \in \mathbb{R}^{d\times d\times d\times d}$ is a symmetric nominal elasticity tensor being of the type of elastic stiffness. Thus, the data-driven paradigm aims to find the closest point $\boldsymbol{z}$ in the constraint set $\mathcal{C}$ to $\hat{z}$ in the material data set $\mathcal{D}$.

Challenges such as data availability, noise, inconsistency, and high dimensionality frequently arise in the data-driven paradigm. Traditional analytical and computational methods may need to be adjusted when addressing these issues. Consequently, the incorporation of machine learning, particularly methods like generative adversarial networks coupled with physics-informed generators, is considered. This integration is aimed at effectively handling the complexities of data-driven datasets, ensuring the outcomes remain consistent with domain-specific knowledge. The following sections will present a detailed discussion on the principles of artificial neural networks and physics-informed neural networks, illustrating the approach of physics-informed generative adversarial networks to solve the data-driven boundary value problem (4.4).

4.3 Generative adversarial networks with physics-informed generators for model-free data-driven problems

This section delves into the application of Generative Adversarial Networks (GANs) equipped with Physics-Informed Generators for addressing the model-free data-driven problem. A GAN involves a competitive dynamic between two neural networks, forming a zero-sum game: one network's success implies the other's setback. To harness GANs for resolving the data-driven boundary value problem depicted in (4.4), Section 4.3.1 initiates with a concise overview of Artificial Neural Networks (ANNs) and explains physics-informed neural networks (PINNs). Section 4.3.2 lays out the foundational principles of GANs, and in Section 4.3.3, we pivot to the novel approach of leveraging GANs augmented with PINNs to solve the data-driven boundary value problem.

4.3.1 Physics-informed neural networks

Based on the universal function approximation theorem [135], an artificial neural network is a parametrized, non-linear function composition that can approximate arbitrary Borel measurable functions. This section introduces the basic concept based on the definitions and formulations in [130]. For this purpose, we introduce a densely connected feed-forward neural network, denoted by the map $\mathcal{N} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$, which is defined by a composition of $n_L \in \mathbb{N}$ non-linear functions:

$$\mathcal{N} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y} \tag{4.9}$$

$$(\mathrm{x}) \mapsto \mathcal{N}(\mathrm{x}) = \boldsymbol{y}^{(\ell)} \circ \ldots \circ \boldsymbol{y}^{(0)} = \boldsymbol{y}, \tag{4.10}$$

for $\ell = 1, \ldots, n_L$, where x denotes the spatial part of the input vector of dimension $d_x \in \mathbb{N}$ and $\boldsymbol{y}$ denotes the output vector of dimension $d_y \in \mathbb{N}$. In this context, $\boldsymbol{y}^{(0)}$ and $\boldsymbol{y}^{(n_L)}$ are called the input and output layer, such that

$$\boldsymbol{y}^{(0)} = \mathrm{x}, \qquad \boldsymbol{y}^{(n_L)} = \mathcal{N}(\mathrm{x}). \tag{4.11}$$

The functions $\boldsymbol{y}^{(\ell)}$ are called hidden layers and define a $\ell-$fold composition, mapping the input x to the output $\boldsymbol{y}$ by

$$\boldsymbol{y}^{(\ell)} = \{\boldsymbol{y}_\eta^{(\ell)},\, \eta = 1, \ldots, \eta_u\}, \text{ with } \boldsymbol{y}_\eta^{(\ell)} = \mathrm{act}^{(\ell)}\left(\mathbf{W}_\eta^{(\ell)} \boldsymbol{y}^{(\ell-1)} + \mathrm{b}_\eta^{(\ell)}\right). \tag{4.12}$$

We call $\boldsymbol{y}_\eta^{(\ell)}$ the η^{th} neural unit of the ℓ^{th} layer $\boldsymbol{y}^{(\ell)}$, where $\eta_u \in \mathbb{N}$ is the total number of neural units per layer. $\mathbf{W}_\eta^{(\ell)}$ and $\mathbf{b}_\eta^{(\ell)}$ denote the weight matrix and bias vector of the η^{th} neural unit in the ℓ^{th} layer $\boldsymbol{y}^{(\ell)}$. Furthermore $\operatorname{act}^{(\ell)}(\cdot) : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function. All weights and biases of all layers $\boldsymbol{y}^{(\ell)}$ are assembled in

$$\boldsymbol{\theta} = \Big\{ \big(\mathbf{W}_\eta^{(\ell)}, \mathbf{b}_\eta^{(\ell)}\big) \,;\, \ell = 1, \ldots, n_L,\, \eta = 1, \ldots, \eta_u \Big\}, \tag{4.13}$$

including all parameters of the network. As a result, the notation $\mathcal{N}(\mathrm{x}; \boldsymbol{\theta})$ highlights the dependence of a neural network's output on the input and the current realization of the weights and biases.

The main idea of solving boundary value problems with an artificial neural network is the reformulation to an optimization problem [136, 78, 137], where the residual of the differential equations is to be minimized. To solve the differential Eqs. (4.1) and (4.2), a suitable topology for the artificial neural network and, consequently, the physics-informed neural networks has to be chosen. Thus, we define the neural network as an ansatz for the displacement and stress field, i.e.

$$\boldsymbol{u}(\mathrm{x}, t) \approx \mathcal{N}_u(\mathrm{x}; \boldsymbol{\theta}_u), \tag{4.14}$$
$$\boldsymbol{\sigma}(\mathrm{x}, t) \approx \mathcal{N}_\sigma(\mathrm{x}; \boldsymbol{\theta}_\sigma), \tag{4.15}$$

with trainable network parameters $\boldsymbol{\theta} := \{\boldsymbol{\theta}_u, \boldsymbol{\theta}_\sigma\}$, where each network utilizes a single set of parameters, $\boldsymbol{\theta}_u$ for displacement and $\boldsymbol{\theta}_\sigma$ for stress. Notably, there is no separate network for the strain tensor. The strain tensor is deduced using the kinematics and differentiation applied to the displacement network, i.e. $\boldsymbol{\varepsilon} = \nabla^{\text{sym}} \mathcal{N}_u(\mathrm{x}; \boldsymbol{\theta}_u)$. In three-dimensional space, defined by the coordinates (x, y, z), the networks outputs are formulated as

$$\begin{aligned} \mathcal{N}_u(\mathrm{x}; \boldsymbol{\theta}_u) &= \{u_i(\mathrm{x}; \boldsymbol{\theta}_u) \,|\, i = x, y, z\}, \\ \mathcal{N}_\sigma(\mathrm{x}; \boldsymbol{\theta}_\sigma) &= \{\sigma_{ij}(\mathrm{x}; \boldsymbol{\theta}_\sigma) \,|\, i, j = x, y, z \text{ and } ij = ji\}, \end{aligned} \tag{4.16}$$

encompassing the three components u_i of displacement $\boldsymbol{u}$ and the six stress components σ_{ij}, with $ij = ji$ ensuring the symmetry of the stress tensor $\boldsymbol{\sigma}$. The topology of the artificial neural network is visualized in Fig.4.3.1.

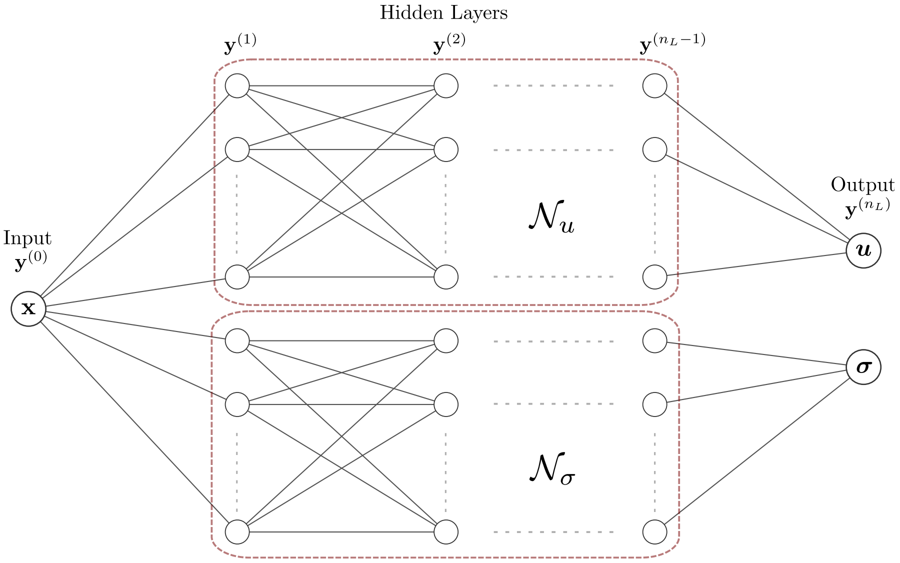


Figure 4.3.1: Schematic representation of the neural network's topology, illustrating the progression from input through multiple hidden layers, resulting in displacement $\boldsymbol{u}$ and stress $\boldsymbol{\sigma}$.

Using the neural network ansatz we can rewrite the physical Eqs. (4.1) and (4.2) as

$$\begin{aligned} R_\Omega &= \nabla \cdot \mathcal{N}_\sigma(\mathbf{x}, \boldsymbol{\theta}_\sigma) - \boldsymbol{f}(\mathbf{x}), && \text{in } \Omega, \\ R_{\Gamma_D} &= \mathcal{N}_u(\mathbf{x}, \boldsymbol{\theta}_u) - \boldsymbol{g}(\mathbf{x}), && \text{on } \Gamma_D, \\ R_{\Gamma_N} &= \mathcal{N}_\sigma(\mathbf{x}, \boldsymbol{\theta}_\sigma) \cdot \boldsymbol{n}(\mathbf{x}) - \boldsymbol{t}(\mathbf{x}), && \text{on } \Gamma_N, \end{aligned} \tag{4.17}$$

where R_Ω penalizes the residual of the equilibrium equation, and the equations R_{Γ_D} and R_{Γ_N} describe the discrepancy of the Dirichlet and Neumann boundary conditions. Notice that if $\mathcal{N}_u$ and $\mathcal{N}_\sigma$ is a solution to the original boundary value problem, they minimize the differential equation-based residuals.

The parameters $\boldsymbol{\theta}$ of the networks can be found by incorporating the physics-induced residuals into the training process of a neural network as components of the loss function. For this, we use a collocation method discretizing the domain Ω and the boundary $\Gamma := \Gamma_D \cup \Gamma_N$ into sets of sample points S_Ω and S_Γ with cardinalities $|S_\Omega|$ and $|S_\Gamma|$. Then, an optimization problem to find the optimal parameters $\boldsymbol{\theta}^\star$, also called training, is defined as

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} L_{\mathcal{C}} \tag{4.18}$$

with $L_{\mathcal{C}} := L_\Omega(\mathbf{x}, \boldsymbol{\theta}) + L_\Gamma(\mathbf{x}, \boldsymbol{\theta})$ given by the local losses

$$L_\Omega = \frac{1}{|S_\Omega|} \sum_{\mathbf{x} \in S_\Omega} \|R_\Omega(\mathbf{x}; \boldsymbol{\theta})\|_2^2, \tag{4.19}$$

$$L_\Gamma = \frac{1}{|S_\Gamma|} \left(\sum_{\mathbf{x} \in S_{\Gamma_D}} \|R_{\Gamma_D}(\mathbf{x}; \boldsymbol{\theta})\|_2^2, + \sum_{\mathbf{x} \in S_{\Gamma_N}} \|R_{\Gamma_N}(\mathbf{x}; \boldsymbol{\theta})\|_2^2 \right). \tag{4.20}$$

The expressions penalize the residual of the governing equations and the discrepancy of the Dirichlet and Neumann boundary conditions, respectively. Fig. 4.3.2 illustrates the complete physics-informed topology.

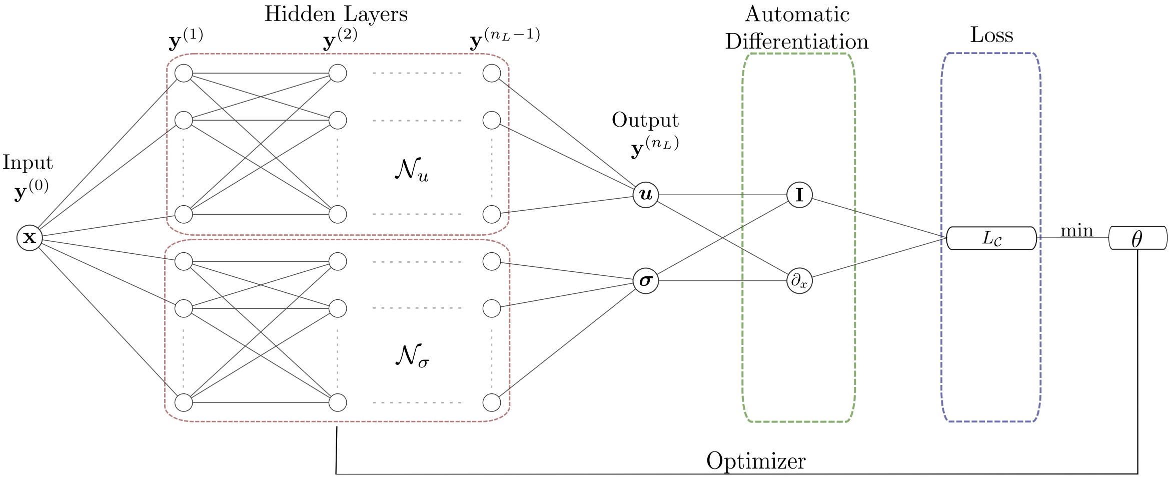


Figure 4.3.2: Schematic representation of the physics-informed neural network's topology, illustrating the progression from input through multiple hidden layers, resulting in displacement $\boldsymbol{u}$ and stress $\boldsymbol{\sigma}$. These outputs undergo automatic differentiation to compute the physics-based loss function $L_{\mathcal{C}}$. The loss is minimized using an optimizer to refine the network parameters θ.

While the PINN framework provides a straightforward method to solve physical-enhanced problems, it has challenges. Notably, there have been instances where the optimization yields solutions with unexpected or non-physical behaviors even when carefully tailored to encapsulate the physics [117]. Additionally, the current PINN formulation must minimize the difference between the network's outputs and the available strain-stress data $\mathcal{D}$ due to the nature of the data-driven distance minimization problem (4.4). If we integrate the distance as an additional loss into the global loss, the whole problem becomes a nested optimization, leading to training challenges. The neural network could optimize in an undesired direction during each training epoch. If the approximated strain-stress point is not accurate, the corresponding data point might be suboptimal concerning the optimization algorithm, further complicating the learning process. To address these challenges, we consider the integration of PINNs with generative adversarial networks. GANs are proficient at generating outputs with the same properties as actual data, providing a potential approach to generating realistic strain-stress solutions. Their flexibility ensures adaptability across diverse data types suited for various physical conditions. Moreover, the inherent capability of GANs to discern and capitalize on intricate patterns may lead to a more robust representation of underlying physics. Additionally, with conditional GANs, generating outputs based on specific conditions becomes feasible, allowing for more targeted solutions. The combined PINN-GAN approach seeks to ensure physical consistency and alignment with observed data, leveraging the strengths of both methodologies. For clarity, we will provide a brief overview of GAN theory in the following.

4.3.2 Intermezzo to generative adversarial networks

Introduced by Goodfellow et al. [21], generative adversarial networks illustrate a novel approach to generating data using neural architectures. These networks comprise two distinctive neural entities: the generator (G) and the discriminator (D). The underlying goal of a GAN is to generate data instances that emulate the properties of actual data. The generation is achieved by setting the two networks against each other in a competitive game, often described as a dual-player minimax game.

Taking reference from the definitions provided in (4.9), we define the real data space as $\mathbb{D}_{\text{real}} \subset \mathbb{R}^{d_y}$, where d_y is the dimension of the space, i.e., $d_y = \dim(\mathbb{D}_{\text{real}})$. The main objective of GANs is to produce synthetic data denoted as $\mathbf{y}_{\text{syn}}$, residing in the same space as our real data $\mathbf{y}_{\text{real}}$. The generator can be defined as a function $G : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$, which transforms a random noise vector $\mathbf{x}$ into synthetic data $\mathbf{y}_{\text{syn}}$. In contrast, the discriminator operates as a function $D : \mathbb{R}^{d_y} \to \mathbb{R}$, that provides a measure of authenticity for a given data sample. Mathematically, these networks can be illustrated as:

$$\begin{aligned} G : \mathbb{R}^{d_x} &\to \mathbb{R}^{d_y} \\ \mathbf{x} &\mapsto \mathcal{N}_G(\mathbf{x}; \boldsymbol{\theta}_G), \end{aligned} \qquad \text{and} \qquad \begin{aligned} D : \mathbb{R}^{d_y} &\to [0,1] \\ \mathbf{y} &\mapsto \mathcal{N}_D(\mathbf{y}; \boldsymbol{\theta}_D). \end{aligned} \tag{4.21}$$

Here, $\mathcal{N}_G(\mathbf{x}; \boldsymbol{\theta}_G)$ and $\mathcal{N}_D(\mathbf{y}; \boldsymbol{\theta}_D)$ describe the neural networks with their corresponding trainable parameters $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$. The adversarial game between the generator and the discriminator during training can be encapsulated in the following objective

$$L(G, D) = \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}}[\ln D(\mathbf{y})] + \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\ln(1 - D(G(\mathbf{x})))], \tag{4.22}$$

leading to the optimization:

$$\min_G \max_D L(G, D), \tag{4.23}$$

where $\mathbb{E}$ represents a random variable's expectation or expected value. It provides a weighted average of a function concerning its probability distribution. Specifically,

$$\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}}[\ln D(\mathbf{y})] \tag{4.24}$$

represents the average logarithmic score assigned by the discriminator to actual data samples drawn from the distribution p_{data}. On the other hand, the expression

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}[\ln(1 - D(G(\mathbf{x})))] \tag{4.25}$$

reflects the average logarithmic score the discriminator accords to the synthetic or generated data, which is created from a random noise vector $\mathbf{x}$ following the noise distribution $p_{\mathbf{x}}$. The competition between the two networks is straightforward: the generator G aims to produce data that the discriminator D cannot distinguish from accurate data. In contrast, the discriminator tries to better distinguish real data from fake data produced by G. The probability distributions p_{data} and $p_{\mathbf{x}}$ depict the actual data and noise distributions, respectively. The terms in the objective function essentially capture the average confidence levels of the discriminator in judging the authenticity of both original and fake data samples. The procedure of the GAN's interplay between the generator and the discriminator is illustrated in Fig. 4.3.3.

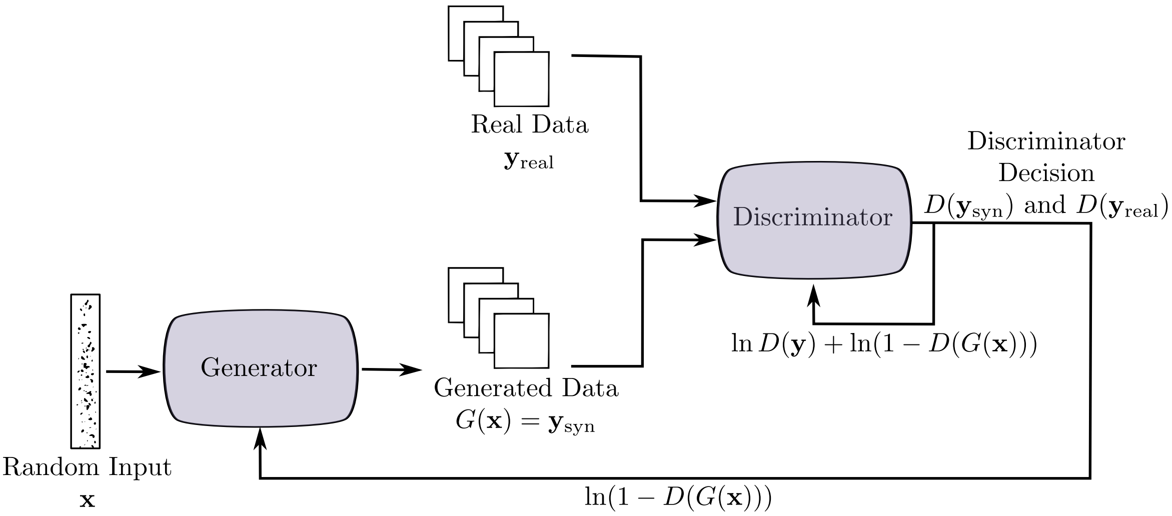

Real Data
$\mathbf{y}_{\text{real}}$
Discriminator
Decision
$D(\mathbf{y}_{\text{syn}})$ and $D(\mathbf{y}_{\text{real}})$
$\ln D(\mathbf{y}) + \ln(1 - D(G(\mathbf{x})))$
Generator
Generated Data
$G(\mathbf{x}) = \mathbf{y}_{\text{syn}}$
Random Input
$\mathbf{x}$
$\ln(1 - D(G(\mathbf{x})))$

Figure 4.3.3: Schematic representation of a generative adversarial network (GAN) showcasing the interaction between the generator producing data from random input and the discriminator evaluating the authenticity of both real and generated data.

Generative Adversarial Networks (GANs) have marked a significant advancement in generative modeling. However, challenges such as mode collapse, training instability, and vanishing gradients often hinder their application. A particularly notable issue is mode collapse, where the generator learns to produce a limited variety of outputs, reducing the model's overall effectiveness. Additionally, traditional GANs employ a sigmoid activation function in the discriminator's final layer, constraining the output to a $[0, 1]$ range. It often leads to loss of values that are difficult to interpret, complicating the assessment of training quality. We propose utilizing the Wasserstein distance to address these challenges, as introduced in [138]. The Wasserstein GAN (WGAN) leverages this distance to provide a more stable training dynamic. The objective function of the WGAN is formulated as:

$$L_{WGAN}(G, D) = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}}[D(\mathbf{y})] - \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}[D(G(\mathbf{x}))], \tag{4.26}$$

leading to the optimization:

$$\min_{G} \max_{D} L_{WGAN}(G, D). \tag{4.27}$$

Building upon the foundation of the WGAN, the Wasserstein GAN with Gradient Penalty (WGAN-GP) introduces a regularization term. This term ensures that the discriminator's gradients remain bounded, as described in [139], and addresses the vanishing gradient problem. The gradient penalty is crucial for enforcing the Lipschitz continuity condition and is defined as:

$$\text{GP} = \mathbb{E}\left[\left(\|\nabla_{\tilde{\mathbf{y}}} D(\tilde{\mathbf{y}})\|_2 - 1\right)^2\right], \tag{4.28}$$

where $\tilde{\mathbf{y}} = \delta \mathbf{y}_{\text{real}} + (1-\delta)\mathbf{y}_{\text{syn}}$, and δ is sampled from a uniform distribution in $[0, 1]$. The optimization for WGAN-GP is thus given by:

$$\min_{G} \max_{D} L_{WGAN}(G, D) + \omega \cdot \text{GP}, \tag{4.29}$$

where $\omega \in \mathbb{R}_+$ is a hyperparameter determining the weight of the gradient penalty in the overall objective [140].

4.3.3 Physics-informed GANs for data-driven mechanics problems

In the classical data-driven computational mechanics paradigm Section 4.2, the objective is to find the closest point $\boldsymbol{z}$ in the constraint set $\mathcal{C}$ to $\hat{\boldsymbol{z}}$ in the material dataset $\mathcal{D}$, as formalized in Eq. (4.4). This context motivates our modified GAN approach for data-driven mechanics problems. To utilize GANs for solving differential equations in a data-driven mechanics setting, we propose a novel approach wherein the generator in the GAN architecture is identified as a physics-informed neural network (PINN). In this paradigm, while the generator outputs plausible solutions adhering to the underlying physics, the discriminator is trained to distinguish between the generator's predictions and actual strain-stress data.

In the conventional GAN setup from Section 4.3.2, the generator G maps the input vector $\mathbf{x}$ into synthetic data, $\mathbf{y}_{\text{syn}}$. Instead of treating $\mathbf{x}$ as a random noise vector, it represents the collocation points $\mathbf{x}$ in the domain S_Ω. Thus, the generator is formalized as a mapping $G : S_\Omega \to (\mathcal{N}_u, \mathcal{N}_\sigma)$, where $\mathcal{N}_u$ and $\mathcal{N}_\sigma$ represents the neural network approximation for the displacement and stress field, respectively. Therefore, the generator can be defined as:

$$G(\mathbf{x}, \boldsymbol{\theta}_G) := (\mathcal{N}_u(\mathbf{x}; \boldsymbol{\theta}_u), \mathcal{N}_\sigma(\mathbf{x}; \boldsymbol{\theta}_\sigma)) \tag{4.30}$$

where $\boldsymbol{\theta}_G := (\boldsymbol{\theta}_u, \boldsymbol{\theta}_\sigma)$ denotes the trainable parameters of the generator network. Building upon the physics-informed aspect, we differentiate $\mathcal{N}_u$ and employ the kinematics equation to obtain the strain $\boldsymbol{\varepsilon}$. Given $\boldsymbol{\varepsilon} = \nabla^{\text{sym}}\mathcal{N}_u$, the generator's output evolves from merely the neural network predictions $\mathcal{N}_u$ and $\mathcal{N}_\sigma$ to the strain-stress pair $\boldsymbol{z} := (\boldsymbol{\varepsilon}(\mathcal{N}_u), \mathcal{N}_\sigma)$.

Once we obtain the strain-stress output from the generator, to stay consistent with the data-driven mechanics' paradigm, we compute the strain-stress data points $\hat{\boldsymbol{z}} \in \mathcal{D}$ closest to the output $\boldsymbol{z}$, which corresponds to:

$$\hat{\boldsymbol{z}} = \arg\min_{\hat{\boldsymbol{z}} \in \mathcal{D}} d(\boldsymbol{z}, \hat{\boldsymbol{z}}), \tag{4.31}$$

with distance (4.7). We then use $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ as synthetic and real data for the discriminator's training. For the discriminator $D(\boldsymbol{y}, \boldsymbol{\theta}_D)$, we establish the mapping $D : \mathbb{R}^{2\bar{d}} \to [0, 1]$, aligning with the conventional GAN framework. To accommodate strain-stress pairs as inputs for the discriminator, we convert a pair into a $2\bar{d}$-vector $\boldsymbol{y}$ with $\bar{d} = \frac{d^2+d}{2}$ by applying Voigt-Notation to both the strain and stress, then merging them into a single vector. Given strain-stress data $\hat{\boldsymbol{z}} \in \mathcal{D}$, it assesses the data's authenticity, furnishing scores to guide the generator's training. With the generator now representing a PINN, the adversarial loss in Eq. (4.22) has to integrate the physics-informed loss $L_{\mathcal{C}}$, derived from the residuals of the governing differential equations:

$$L(G, D) = \mathbb{E}_{\hat{\boldsymbol{z}} \sim p_{\mathcal{D}}}[\ln D(\hat{\boldsymbol{z}})] + \mathbb{E}_{\mathbf{x} \sim p_{S_\Omega}}[\ln(1 - D(G(\mathbf{x}))) + L_{\mathcal{C}}]. \tag{4.32}$$

The collaborative training between the discriminator and the physics-informed generator ensures that the latter learns to craft data that confounds the discriminator and aligns closely with intrinsic physics. Fig. 4.3.4 illustrated the physics-enhanced GAN approach for the data-driven mechanics problem. Regarding Wasserstein GANs and their gradient penalty variants, their objectives concerning the physics-informed generator must be modified. For instance, with the Wasserstein GAN objective, the loss function becomes:

$$L_{\text{WGAN}}(G, D) = \mathbb{E}_{\hat{\boldsymbol{z}} \sim p_{\mathcal{D}}}[D(\hat{\boldsymbol{z}})] - \mathbb{E}_{\mathbf{x} \sim p_{S_\Omega}}[D(G(\mathbf{x})) + L_{\mathcal{C}}], \tag{4.33}$$

Moreover, for the WGAN-GP, the combined objective is:

$$L(G, D) = L_{\text{WGAN}} + \omega \cdot \text{GP}. \tag{4.34}$$

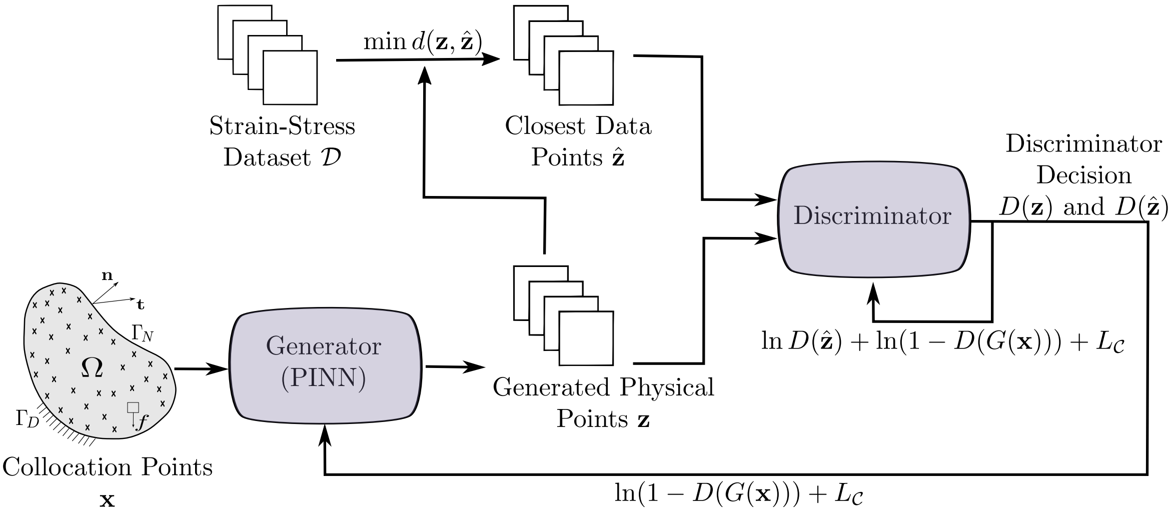

$\min d(\mathbf{z}, \hat{\mathbf{z}})$
Strain-Stress Dataset $\mathcal{D}$
Closest Data Points $\hat{\mathbf{z}}$
Discriminator Decision $D(\mathbf{z})$ and $D(\hat{\mathbf{z}})$
Discriminator
$\mathbf{n}$
$\mathbf{t}$
Γ_N
Ω
Γ_D
f
Generator (PINN)
Generated Physical Points $\mathbf{z}$
$\ln D(\hat{\mathbf{z}}) + \ln(1 - D(G(\mathbf{x}))) + L_{\mathcal{C}}$
Collocation Points $\mathbf{x}$
$\ln(1 - D(G(\mathbf{x}))) + L_{\mathcal{C}}$

Figure 4.3.4: Schematic representation of a physics-informed generative adversarial network (GAN) incorporating collocation points and strain-stress data for physical point generation and discrimination.

By incorporating GANs with physics-informed principles, the models produce data that adheres to the statistics of observed datasets and the underlying differential equations. This integration addresses the nested optimization issue commonly found in the PINN-based data-driven mechanics. With the capability of GANs to generate outputs mirroring accurate data, the solutions are both statistically relevant and in line with physical principles. Using GANs simplifies the optimization process, making the training more stable and less prone to errors from inaccurate strain-stress approximations. However, it is worth noting that the loss values obtained while training a traditional GAN often need to be revised. In many studies, qualitative and quantitative evaluation methods are employed to assess the performance of the GAN. Qualitative evaluations, while offering a quick visual validation, can be subjective. Typically, they involve human observers who evaluate the realism of a generated sample. The overall presumption has been that if the generated sample appears realistic, the GAN's training is deemed successful, regardless of potential fluctuations in loss values. Nevertheless, such evaluations can be biased and do not always represent the complete performance spectrum of the GAN. Considering these challenges, especially in the context of our work where the goal is not generating images but accurately representing strain-stress states, we decided on WGAN + GP. Unlike traditional GANs, the loss of WGANs has a convergence point. Ideally, this point is reached when the generator is so adept at producing samples that no Lipschitz continuous discriminator can differentiate between real and generated samples. This characteristic of WGAN provides a more stable and consistent evaluation metric, ensuring that the generated strain-stress states are physically accurate. The effectiveness of this method will be showcased in a two-dimensional numerical example.

4.4 Numerical benchmark of a non-linear elastic plate with hole

This section illustrates the application of GANs to the data-driven computing paradigm [30] in a typical benchmark, considering stress analysis of non-linear elastic material. We discuss the problem setup and test environments and give a proper definition of the geometry and boundary conditions and the material parameters for data generation. We limit the simulation to noiseless synthetic data sets, which consist of strain-stress points created numerically using a material model rather than obtained by actual experimental measurements. However, experimental data is often noisy and contains outliers. This issue can be addressed with noise reduction algorithms such as tensor voting [105], Kalman filtering [75], and deep learning-based techniques.

This benchmark investigates a two-dimensional in-plane plate with a hole subject to a distributed force. The geometry, boundary conditions, and displacements are chosen according to a similar test presented in [48] and illustrated in Fig. 4.4.1.

Geometry:

The system is defined by $\Omega = \left[-\frac{\ell}{2}, \frac{\ell}{2}\right]^2 \setminus B_r(0)$, where B_r refers to the open ball of radius $r = \frac{\ell}{4}$ centered at the origin $(0, 0)$. The side lengths of the plate are equal to $\ell = 2\text{m}$. Due to the symmetry of geometry, only one-quarter of the system is simulated, cf. Fig 4.4.1. Displacements are fixed at the quarter plate's left surface $x = 0$ in x-direction and at the bottom surface $y = 0$ in y-direction. The corresponding conditions read as follows:

$$\begin{cases} u_x = 0, & \text{if } x = 0; \\ u_y = 0, & \text{if } y = 0; \end{cases} \tag{4.35}$$

where u_x and u_y are the displacements in x and y-directions, respectively. In addition, we define boundary conditions for the stress, especially for $x = \frac{\ell}{2}$ the plate is subjected to a distributed force $t(y) = 200\cos(\frac{\pi y}{2})$ in x-direction. The boundary conditions for the stress

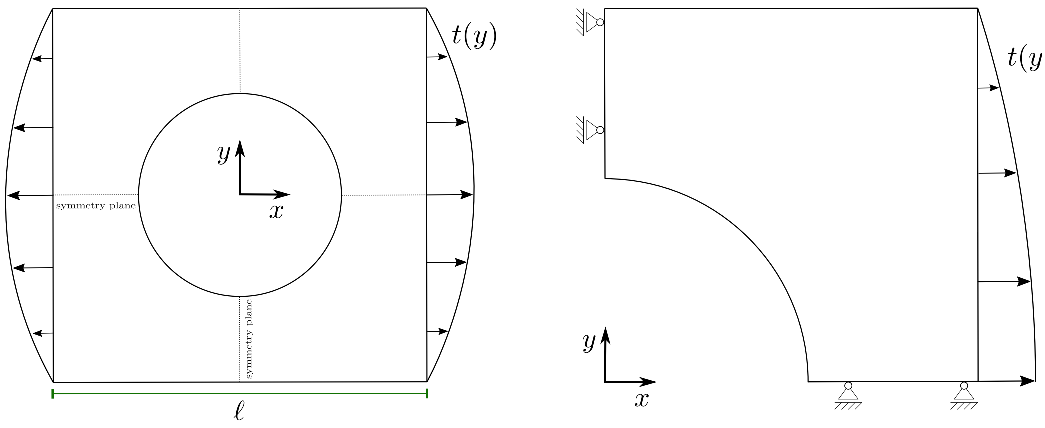


Figure 4.4.1: Illustration of a square plate subjected to external forces, alongside its top-right quadrant representing the symmetry section with specified boundary conditions and force distribution $t(y)$ applied.

components read

$$\begin{cases} \sigma_{xx} = t(y), & \text{if } x = 1; \\ \sigma_{yy} = 0, & \text{if } y = 1; \\ \sigma_{xy} = 0, & \text{if } x \in \{0,1\} \vee y \in \{0,1\}, \end{cases} \tag{4.36}$$

where σ_{xx}, σ_{yy} and σ_{xy} are the stress components in x and y-directions, respectively. Notice that numerical methods based on the weak form of a boundary value problem innately satisfy stress-free boundary conditions on free boundaries. However, our PINN approach utilizes the strong form of the boundary value problem, so it is crucial to impose the zero stress boundary conditions directly [130].

Material parameters:

The boundary value problem considers the non-linear elastic material behavior of [48] defined by

$$\boldsymbol{\sigma} = \lambda g(\operatorname{tr}(\boldsymbol{\varepsilon}))\boldsymbol{I} + \mu\boldsymbol{\varepsilon} + \boldsymbol{C}\boldsymbol{\varepsilon}, \tag{4.37}$$

with $g(x) = ((|x| + a)^p - a^p)\operatorname{sgn}(x)$ and $a, p \in \mathbb{R}$. The applied material parameters are Young's modulus E, Poisson's ratio ν, and orthotropic elasticity tensor for plane strain given by

$$\boldsymbol{C} = \begin{pmatrix} C_{11} & 2\nu(\bar{\lambda} + G_\perp) & 0 \\ 2\nu(\bar{\lambda} + G_\perp) & \bar{\lambda} + 2G_\perp & 0 \\ 0 & 0 & G_\parallel \end{pmatrix}, \tag{4.38}$$

where $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$ are the well known Lamé constants and $C_{11} = 4.6875E$, $G_\perp = 0.3E$, $G_\parallel = 0.2E$ and $\bar{\lambda} = \frac{2\nu^2+1}{15-20\nu^2}E$ are additional material parameters. The exact parameter values used for the reference solution and synthetic data are given in Table 4.4.1.

$E\,[\text{MPa}]$	$\nu\,[-]$	$a\,[-]$	$p\,[-]$
1×10^4	0.3	0.001	0.005

Table 4.4.1: Material parameters

Synthetic data:

To simulate actual experimental measurements, we generate data artificially using the nonlinear material model (4.37) based on the given material parameters. We investigate normal data distributions of 100^3 strain-stress data points with a fixed random seed. The data is created by a zero-mean normal distribution with a standard deviation of 0.015 in all strain dimensions. To compare the model's behavior under measurement uncertainties, we introduce Gaussian noise to the dataset, with a maximum amplitude of 5%.

WGAN parameter:

Our study utilized the WGAN-GP framework to simulate the boundary value problem through a data-driven approach. To fine-tune our network's performance, we optimized hyperparameters, which involved exploring a range of network configurations to identify the most effective setup, including the number of layers, neuron density, and activation functions. The network is structured with a generator and discriminator setup as follows.
The generator is defined as a tuple of two neural networks by

$$G(\mathrm{x},\boldsymbol{\theta}_G)=\{\mathcal{N}_u(\mathrm{x},\boldsymbol{\theta}_u),\mathcal{N}_\sigma(\mathrm{x},\boldsymbol{\theta}_\sigma)\},\tag{4.39}$$

$\mathrm{x}=(x,y)$ and $\boldsymbol{\theta}_G=(\boldsymbol{\theta}_u,\boldsymbol{\theta}_\sigma)$ are the network parameters regarding the displacement and stress component, respectively. Following (4.16) the output components are given by

$$\begin{aligned}\mathcal{N}_u(\mathrm{x},\boldsymbol{\theta}_u)&=\{u_i(\mathrm{x},\boldsymbol{\theta}_u)\,|\,i=x,y\},\\ \mathcal{N}_\sigma(\mathrm{x},\boldsymbol{\theta}_\sigma)&=\{\sigma_{ij}(\mathrm{x},\boldsymbol{\theta}_\sigma)\,|\,i,j=x,y\text{ and }ij=ji\}.\end{aligned}\tag{4.40}$$

The networks are constructed with a series of fully connected layers. The architecture utilizes 3 hidden layers, each with 128 neurons. The activation function used across these layers is the Swish function, defined as

$$\text{Hardswish}(x)=\begin{cases}0 & \text{if } x\leq -3,\\ x & \text{if } x\geq 3,\\ \frac{x^2+3x}{6} & \text{otherwise.}\end{cases}\tag{4.41}$$

In addition, to optimize the network training, we hard enforce the boundary conditions from (4.35) and (4.36), such that the output of the generator is given by

$$\begin{aligned}u_x(\mathrm{x};\boldsymbol{\theta}_u)&=x\cdot\hat{u}_x(\mathrm{x};\boldsymbol{\theta}_u),\\ u_y(\mathrm{x};\boldsymbol{\theta}_u)&=y\cdot\hat{u}_y(\mathrm{x};\boldsymbol{\theta}_u),\\ \sigma_{xx}(\mathrm{x};\boldsymbol{\theta}_\sigma)&=x\cdot t(y)+(1-x)\cdot\hat{\sigma}_{xx}(\mathrm{x};\boldsymbol{\theta}_\sigma),\\ \sigma_{yy}(\mathrm{x};\boldsymbol{\theta}_\sigma)&=(1-y)\cdot\hat{\sigma}_{yy}(\mathrm{x};\boldsymbol{\theta}_\sigma),\\ \sigma_{xy}(\mathrm{x};\boldsymbol{\theta}_\sigma)&=xy(1-x)(1-y)\cdot\hat{\sigma}_{xy}(\mathrm{x};\boldsymbol{\theta}_\sigma).\end{aligned}\tag{4.42}$$

On the other hand, the discriminator $D(\boldsymbol{y}, \boldsymbol{\theta}_D)$ comprises a network architecture of 3 hidden layers with $128, 64$ and 32 neurons, which use the LeakyReLU activation function defined as

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha x & \text{if } x < 0, \end{cases} \tag{4.43}$$

with a slope of $\alpha = 0.2$ for negative values.

Training and Validation:

We employ a dataset comprising 128^2 quasi-random points generated using the Sobol sequence [141] for the network's training. It ensures a comprehensive and evenly distributed set of training data points across Ω. For the optimization process, both the generator and discriminator utilize the ADAM optimizer, with beta values set to $(0.5, 0.999)$. Additionally, a learning rate scheduler is integrated to optimize the training process. The scheduler operates with a maximum learning rate of 0.02, adjusting dynamically throughout 200 epochs for both the generator and the discriminator. The computation of spatial derivatives for obtaining the strains and optimizing the loss function is achieved through automatic differentiation. A separate set of 256^2 domain points is generated using a uniform random distribution to validate the network's training. The validation process involves comparing the network's output with the results obtained from a standard Finite Element Method (FEM) simulation. This simulation, embedded within a Newton-Raphson algorithm, calculates the displacement $\boldsymbol{u}_{\text{ref}}$ and stress distribution $\boldsymbol{\sigma}_{\text{ref}}$ across a finite mesh of the domain. The solution fields at the validation points are then interpolated to compare against the network's output. To quantify the global relative difference, we define the error function using the L_2-norm, i.e.

$$\frac{||\boldsymbol{z} - \boldsymbol{z}_{\text{ref}}||_{L_2(\Omega)}}{||\boldsymbol{z}_{\text{ref}}||_{L_2(\Omega)}}, \tag{4.44}$$

where $\boldsymbol{z}$ and $\boldsymbol{z}_{\text{ref}}$ represent the approximation and the reference fields, respectively.

Result:

Our analysis, depicted in Fig. 4.4.2, showcases the strain-stress distribution achieved after training the WGAN-GP model. This analysis is part of our broader objective to compare the WGAN-GP model's performance with a hyper-parameter-optimized PINN model, especially in terms of accuracy against the reference solution. Fig. 4.4.3 displays these losses for both the discriminator and generator, highlighting the balance between the two networks to avoid stagnation in training. Notably, shallow loss values for either the generator or the discriminator can be counterproductive. It generally indicates that one network is dominating the other, leading to stagnation in the training process. Ideally, there should be a balance where both networks challenge each other, encouraging continuous improvement. The distance between generated states and the dataset $\mathcal{D}$ is further demonstrated in Fig. 4.4.4, indicating convergence. Given the data-driven approach, the learning process results in losses converging to a positive lower bound rather than zero. Notice, due to batch processing, the number of training steps exceeds the epoch count.

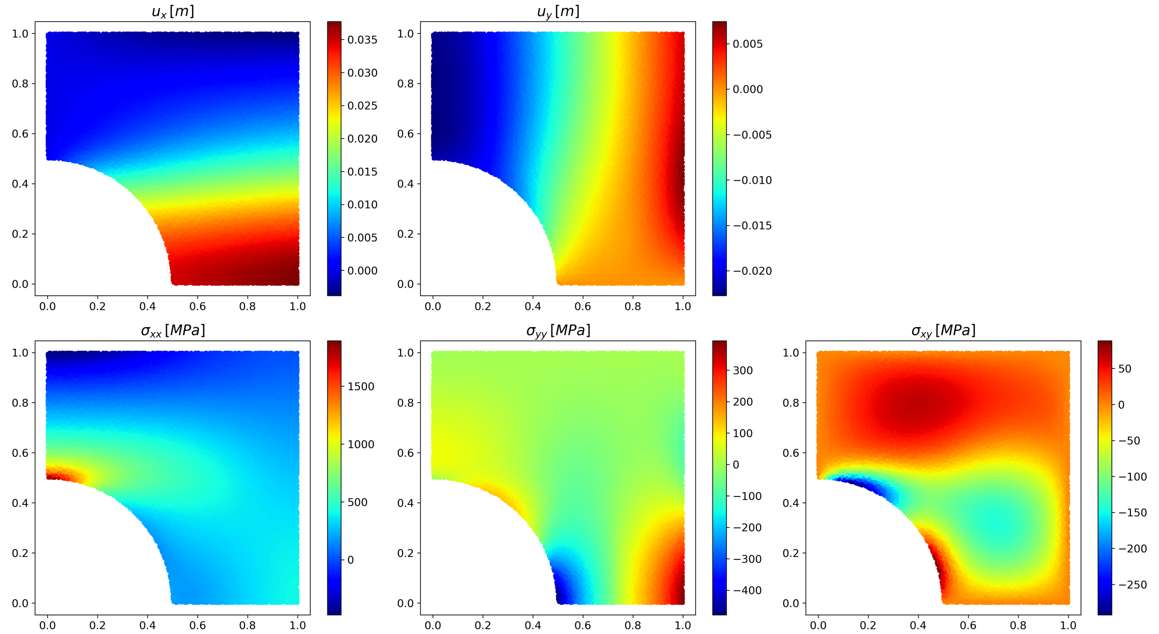


Figure 4.4.2: Visualization of displacement and stress distribution after 200 training epochs offering insights into the material's behavior under the applied loads and conditions (4.35) and (4.36). From top-left to bottom-right: u_x showcases a gradient, indicating a maximum displacement at $(x, y) = (1, 0)$; u_y reveals a displacement trend with negative values highlighted in $x = 0$; σ_x shows a maximum stress magnitude at $x = 0$; σ_y displays a similar gradient at $y = 0$; and σ_{xy} captures a pronounced shear stress distribution inside the plate.

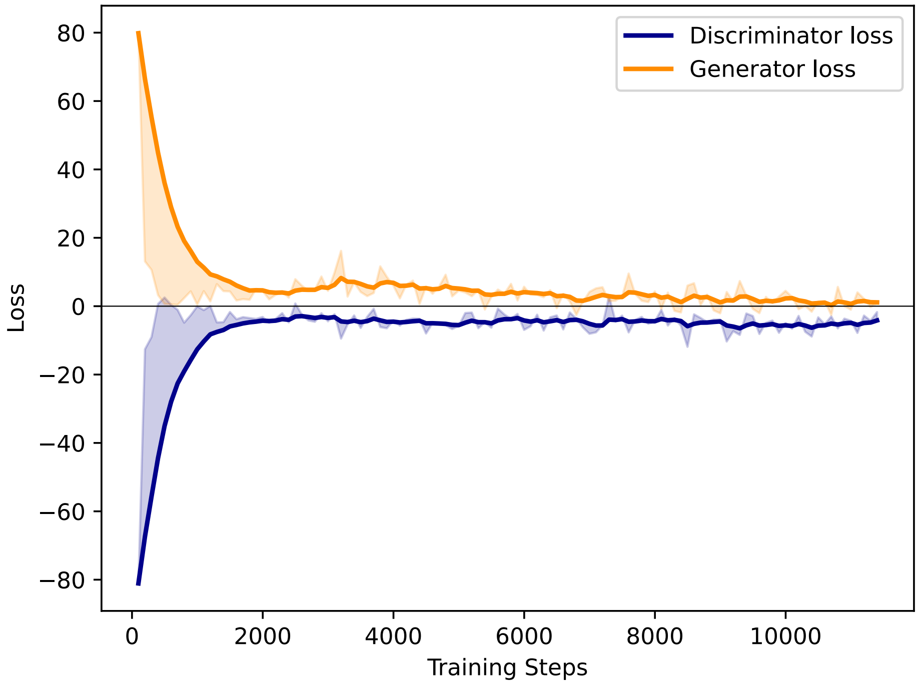


Figure 4.4.3: Comparative Visualization of the discriminator and generator loss metrics over training iterations for a WGAN-GP model, showcasing the dynamic interplay and convergence patterns. The shaded area shows the maximum range of loss for individual training batches.

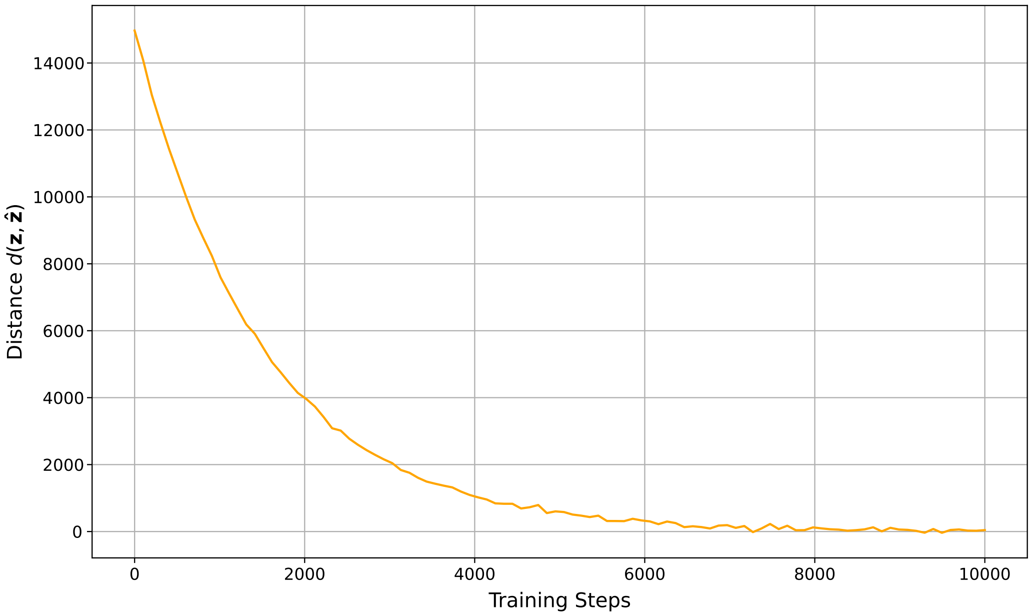


Figure 4.4.4: Visualization of the distance metrics over 200 epochs. The distance of the generated z to the dataset $\mathcal{D}$ illustrates how closely the model-generated outputs match the dataset over training iterations.

In Fig. 4.4.5, we present the field components obtained through the application of the WGAN-GP, PINN, and FEM methods for $x = y$ within the domain Ω. Notably, the results suggest that the WGAN-GP approach provides a more suitable approximation to the solution of the boundary value problem, highlighted by the error analysis. The error involved both local and global estimates to evaluate the method's effectiveness. This approach captures the magnitude of discrepancies through absolute errors while investigating these differences through global relative errors. By integrating these metrics, we avoid the potential distortion caused by solely focusing on one type of error, particularly in scenarios involving small numerical values.
Each field component was individually assessed for the local absolute error, as shown in Fig. 4.4.6. In comparison, Fig. 4.4.7 visualizes the PINN's absolute errors. The component-wise error analysis provides insight into both models' accuracy at the validation points, emphasizing its precision in resolving specific field components. Even though the GAN approach performs better than the PINN approach, there are still regions of more significant errors. A reason for that is the limited data size, which can be mitigated by increasing the dataset size and reducing the peak noise within the dataset. Such adjustments are anticipated to accelerate the data-driven method and, consequently, the precision of the error analysis [107].

In addition, we evaluated the global relative error using (4.44) for the solution fields, reflecting the models' overall accuracy. The comparison in terms of the relative error to the FEM solution is presented in Fig. 4.4.8, indicating that the WGAN-GP approach outperforms the PINN method in terms of accuracy and efficiency by achieving a minimum error that is two orders of magnitude lower than that of the PINN model. Furthermore, the WGAN-GP model displays a much faster convergence rate, reaching its minimum error after approximately 200 training epochs. In contrast, the PINN model requires one to two orders of magnitude more training epochs to converge.

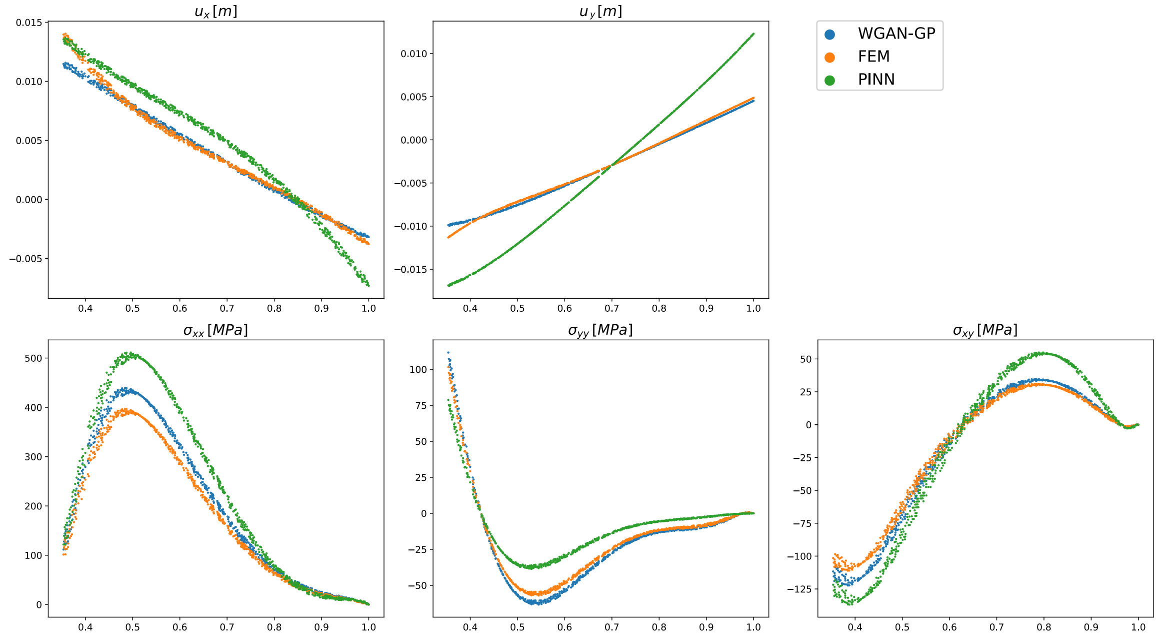


Figure 4.4.5: Comparison of the displacements (u_x, u_y) and stresses $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$ obtained by FEM, WGAN-GP, and PINN along the line $x = y$ within Ω.

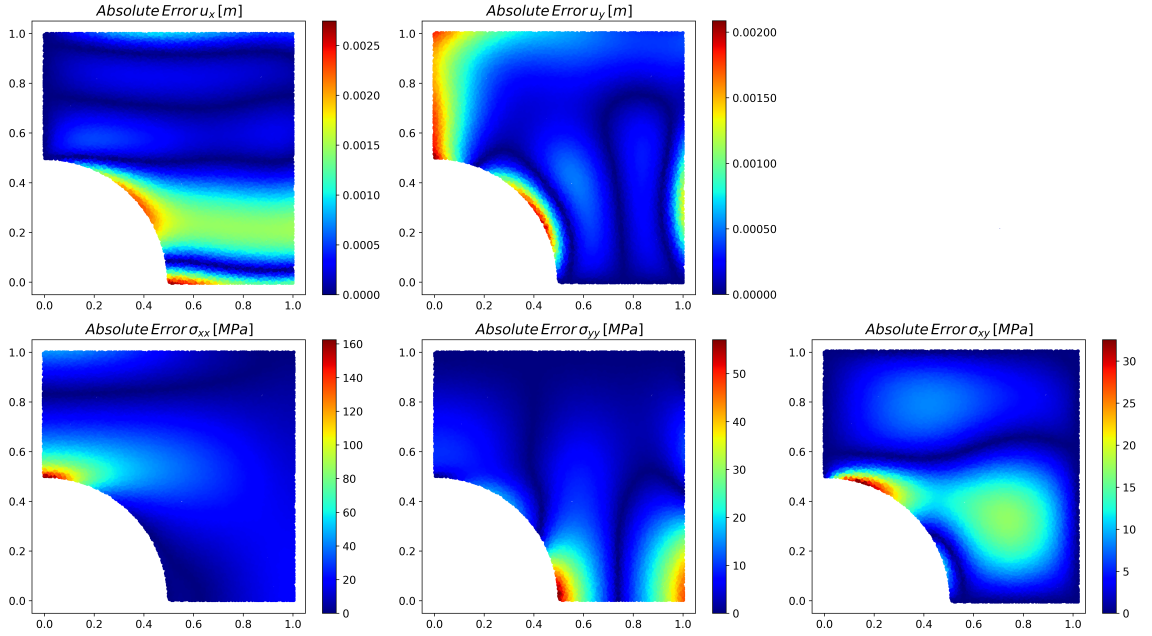


Figure 4.4.6: Visualization of the absolute errors of displacement and stresses in x, y-direction of the WGAN-GP solver compared to the FEM reference solution.

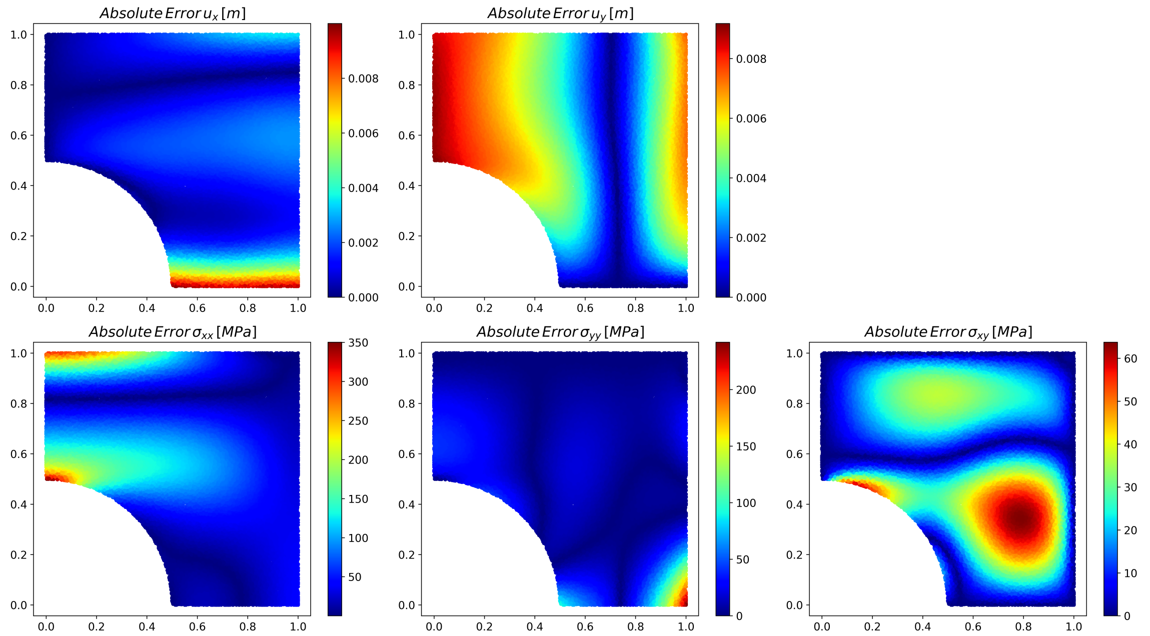


Figure 4.4.7: Absolute error visualization for displacement and stresses in x, y-directions of the PINN solver, compared to the FEM reference.

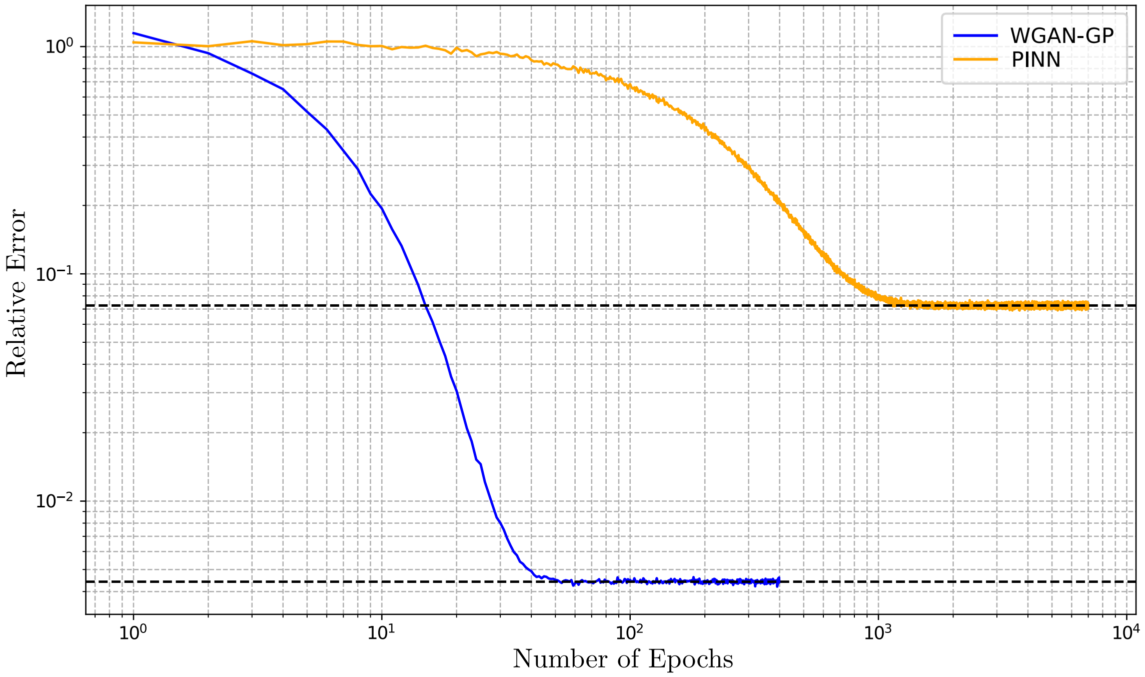


Figure 4.4.8: Global relative error comparison between the WGAN-GP and the standard PINN approach across training epochs. Optimized through hyper-parameter tuning, both models demonstrate convergence towards a lower error boundary.

4.5 Conclusion

The model-free data-driven method, developed by Kirchdoerfer and Ortiz, uses experimental data directly in simulations, bypassing the entire material modeling step. The paradigm uses nearest-neighbor clustering to reformulate boundary value problems. The approach has been diversified for many applications. Challenges such as data availability, noise, inconsistency, and high dimensionality frequently arise in the data-driven paradigm. Traditional analytical and computational methods may need to be adjusted when addressing these issues. Consequently, the incorporation of machine learning methods is considered, especially physics-informed neural networks. In solving boundary value problems with ANNs, the idea is to transform it into an optimization problem. The residual of the differential equations is minimized, and the neural network approximates the displacement and stress field. However, there are challenges with PINNs. There have been instances where the optimization yields solutions with unexpected or non-physical behaviors even when carefully tailored to encapsulate the physics. If we integrate the distance as an additional loss into the global loss, the whole problem becomes a nested optimization, leading to training challenges. In addition, approximated strain-stress fields can correspond to suboptimal data points influencing the direction and rate of the convergence.

To address these challenges, we consider the integration of PINNs with generative adversarial networks. GANs are proficient at generating outputs with the same properties as actual data, providing a potential approach to generating realistic strain-stress solutions. Their flexibility ensures adaptability across diverse data types suited for various physical conditions. Moreover, the inherent capability of GANs to distinguish and capitalize on intricate patterns may lead to a more robust representation of underlying physics. The combined PINN-GAN approach seeks to ensure physical consistency and alignment with observed data, leveraging the strengths of both methodologies.

This research introduced an approach to WGANs-GP tailored for data-driven mechanics problems. The generator is identified as a PINN, ensuring generated outputs conform to underlying physical principles. Instead of random noise, the generator utilizes collocation points from the domain and maps them to neural network approximations of strain and stress fields. The discriminator is then trained using the generated and the closest actual strain-stress data. By integrating WGANs with physics-informed principles, the model outputs adhere to observed dataset statistics and differential equations. This results in improved optimization, more stable training, and accurate, physically consistent solutions. In this regard, we investigated a non-linear elastic plate with a hole benchmark. The results indicate that our proposed method provides reasonable outcomes. Furthermore, we observed robust and consistent training of the networks and noted the convergence of the data-driven solution as data size increased.

As we advance our research, we aim to delve deeper into other convergence criteria for the GAN or WGAN. We plan to explore metrics such as the Inception Score [142], Frechet Inception Distance [143], and perceptual similarity measures [144] to provide a broader assessment of the generated outputs. These metrics will not only help to analyze the quality of the generated material states but could also allow dealing with inelastic and path-dependent material behavior effectively. Another area of interest is using the discriminator in the GAN framework for material identification. The discriminator's ability to distinguish between actual and generated outputs can be used to identify different material states. This approach could offer a novelty to classify materials, and we want to explore this further.

5 Conclusion & Outlook

This dissertation investigated a novel data-driven computing paradigm that bypasses traditional material models, eliminating information loss associated with them. Building upon the initial framework proposed by Kirchdoerfer and Ortiz [30], we have significantly expanded this approach through three distinct methodologies.

Extension to inelasticity: Our first paper addressed the limitations of applying data-driven mechanics to materials exhibiting inelastic behavior, where the material response depends on past deformations. The challenge is that the same stress-strain point could represent different material behaviors based on this history. Our solution involved enriching the data set with directions in the tangent space and categorizing the data into subsets based on these behaviors. By introducing transition rules, we effectively applied our approach to non-linear and elasto-plasticity with isotropic hardening.

Data accessibility and efficiency in inelasticity: Recognizing the importance of data coverage and efficiency in practical applications, the second paper expanded the data-driven computational mechanics approach to address the challenges of obtaining qualified data states and calculating tangent spaces. We focused on reducing the required data for elasto-plasticity with isotropic hardening by exploiting material symmetry and formulating the approach using Haigh-Westergaard coordinates. This approach simplified the data acquisition and leveraged the material's inherent properties to minimize data needs. Due to the Haigh-Westergaard space, combining tension-torsion and single tensile tests was sufficient to cover the needed data, effectively capturing the yield surface and tangent space information. Our adapted data-driven paradigm achieved the same results as our first paper but with significantly less data and computational time.

Integration of machine learning: Our third study aimed to combine the model-free data-driven paradigm with generative adversarial networks (GANs) capabilities. This integration enhances traditional physics-informed neural networks by embedding the principles of data-driven computational mechanics directly within the GAN framework. In the refined model, the generator is constrained by physical principles, i.e., PINNs, ensuring that generated outcomes remain physically admissible. Meanwhile, the discriminator, leveraging the closest available strain-stress data, evaluates the fidelity of the generator's output. This synthesis offers a new method to combine the strengths of data-driven mechanics with deep learning. It provides a more robust and consistent network training, thus a more reliable and accurate method for simulating and predicting complex mechanical behaviors using machine learning.

In summary, we enhanced the capabilities of the model-free data-driven paradigm and integrated machine learning techniques. The paradigm of data-driven computational mechanics presents a unique opportunity to leverage existing and emerging data processing techniques designed to handle large datasets. While machine learning has already proven its effectiveness, several exciting areas remain for future research:

Integration of Advanced Data-Driven Techniques: Recently developed machine learning algorithms hold immense promise for modeling complex material behaviors, particularly those with intricate microstructures or history-dependent responses. Inspired by the biological brain, one promising approach is the Spike Neural Network, which processes information through short voltage pulses and offers potential advantages in capturing the spatiotemporal dynamics of complex material behavior. Reinforcement Learning allows algorithms to learn through trial and error, making it well-suited for optimizing material properties or uncovering hidden relationships within material datasets. Attention Mechanisms focus on specific, relevant parts of the data, potentially leading to more accurate predictions for materials with intricate microstructures or long-range dependencies. Another promising field is the exploration of quantum computing in data-driven computational mechanics.

Quantum Computing: Another promising area is quantum algorithms, which can enhance the processing of large datasets and complex simulations by leveraging the principles of superposition and entanglement. Since Quantum algorithms are tailored for data analysis tasks like classification, clustering, and pattern recognition, future research could investigate how quantum computing can be integrated into the data-driven paradigm to solve boundary value problems more efficiently, particularly for materials with intricate, non-linear behaviors. This research could lead to the discovery of new materials and designs by enabling the rapid exploration of vast material parameter spaces.

List of Figures

1 Introduction

1.1.1 Deformation states of an infinitesimal volume element concerning strain compatibility. (a) Undeformed state where the volume element retains its original geometry. (b) Compatible deformation: showing uniform deformation and preserving continuity within the volume element. (c) Incompatible deformation: showing nonuniform deformation leading to discontinuities and potential internal stresses within the volume element. 2

1.1.2 Illustration of the state of stress on an infinitesimal element within a continuum, demonstrating the conditions for static equilibrium by balancing the internal stresses. 3

1.1.3 Representation of the strain-stress relationship. The material model $\mathcal{E}$ (red, $-$) outlines the theoretical relationship based on the constitutive equation. The material data set $\mathcal{D}$ (black, $\cdot$) illustrates strain-stress measurements. The visualization emphasizes how the theoretical model is informed by, but distinct from, the experimental data 4

1.2.1 Visualization of the mappings $P_\mathcal{C}$ and $P_\mathcal{E}$: (a) demonstrating the projection of a point $\hat{\boldsymbol{z}}$ in the material model $\mathcal{E}$ onto the nearest point $\boldsymbol{z}$ within the constraint set $\mathcal{C}$. (b) showcasing the reverse mapping from the constraint set to the material set, identifying the nearest material state $\hat{\boldsymbol{z}}$ to a given constraint state $\boldsymbol{z}$. 7

1.2.2 Visualization of the fixed-point approach for the relaxed solution scheme (1.7). Iteration between projections: $P_\mathcal{C}$, which maps data points $\hat{\boldsymbol{z}} \in \mathcal{E}$ to the closest constrained points $\boldsymbol{z} \in \mathcal{C}$, and the projection $P_\mathcal{E}$, which locates closest point in the material set to the constrained state provided by $P_\mathcal{C}$. The iteration is visualized by $\{(\hat{\boldsymbol{z}}^k, \boldsymbol{z}^k), (\hat{\boldsymbol{z}}^{k+1}, \boldsymbol{z}^{k+1}), \ldots\}$, converging to minimum distance d_{min}. 7

1.2.3 Visualization of the fixed-point approach for the data-driven relaxed solution scheme (1.11), adapted to minimize the distance to dataset $\mathcal{D}$. Iteration between projections: $P_\mathcal{C}$, which maps data points $\hat{\boldsymbol{z}} \in \mathcal{D}$ to the closest constrained points $\boldsymbol{z} \in \mathcal{C}$, and the projection $P_\mathcal{D}$, which locates the closest point in dataset $\mathcal{D}$ to the state provided by $P_\mathcal{C}$. The iterative process is depicted by the sequence of points $\{(\hat{\boldsymbol{z}}^k, \boldsymbol{z}^k), (\hat{\boldsymbol{z}}^{k+1}, \boldsymbol{z}^{k+1}), \ldots\}$, converging to minimize the distance d_{min}. 9

1.3.1 The distance minimization algorithm iterates between two steps: first, local projection $P_\mathcal{D}$ maps the material point $\boldsymbol{z}^k$ from the conservation manifold $\mathcal{C}$ onto the closest data point $\hat{\boldsymbol{z}}^{k+1}$ within the data set $\mathcal{D}$. Projection $P_\mathcal{C}$ maps the point back onto the resulting equilibrium state $\boldsymbol{z}^{k+1}$ in $\mathcal{C}$, incorporating the associated tangential space $T_\mathcal{D}$ to ensure that the change in material state between loading steps aligns with the data tangent $\Delta\boldsymbol{z}^{k+1}$. 16

1.3.2 Visualisation of stress $\boldsymbol{\sigma}$ and its deviatoric part $\boldsymbol{s}$ in the Haigh-Westergaard stress space. The coordinates are defined by the invariants (ξ, ρ, θ), which depend on the principal stresses $(\sigma_1, \sigma_2, \sigma_3)$. Figure from [80]. 17

1.3.3 The GAN Framework adapted for model-free data-driven methods, highlighting the integration of physically informed generation and data-driven validation. 18

2 Article 1: Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules

2.3.1 Visualization of data-driven method extended by tangent space. Modeling points $\boldsymbol{z}_e^{k+1}$ minimize distance to the tangent space associated with data points $\hat{\boldsymbol{z}}_e^k$, respecting compatibility and equilibrium constraints. Data points $\hat{\boldsymbol{z}}_e^{k+1}$ minimize distance to modeling points $\boldsymbol{z}_e^{k+1}$. Iterations are repeated until the local data assignments remain unchanged or the global distance is less than a certain tolerance. 25

2.4.1 Discretization and boundary conditions for a cylindrical tube under internal pressure. 28

2.4.2 RMSD Error of data-driven solver for normal and uniform distributed data points. (a) Convergence with respect to data size. (b) Dependency of the error on applied noising for a data set of size 16^3 with normal and uniform distribution. The shaded areas show the spread of the error arising from the different data set realizations. 29

2.4.3 Comparison of stress components σ_{xx} (a) and (b), σ_{yy} (c) and (d) and σ_{xy} (e) and (f) in $[Pa]$ between the reference model (left) and the data-driven algorithm (right) based on a normal distribution of size 16^3. 30

2.4.4 Discretization and boundary conditions for a rectangular plate with a circular hole under loading. 31

2.4.5 Geometry and discretization of an example virtual test of a plate with random holes to generate suitable data sets. 32

2.4.6 Von Mises stress distribution at maximum loading at each Gaussian integration point using (a) J_2-plasticity model and (b) data-driven algorithm. . . . 32

2.4.7 Convergence property of the extended data-driven method using transition rules for elasto-plastic material behaviour. (a) Maximum displacement (vertical displacement of lower right vertex versus traction (resultant load of right edge) for different data resolution. (b) RMSD Error for each data resolution. The shaded area shows the deviation of the error arising from different independent virtual tests. 33

3 Article 2: Model-free data-driven inelasticity in Haigh-Westergaard space – a study how to obtain data points from measurements

3.3.1 Visualisation of a stress tensor $\boldsymbol{\sigma}$ and its deviatoric part in the Haigh-Westergaard stress space. 43

3.3.2 Schematic illustration of a normal vector $\hat{\boldsymbol{N}}$ at a random point $\boldsymbol{s}$ of a parametrized yield surface Φ in Haigh-Westergaard coordinates 43

3.3.3 Schematic illustration of data points $(\hat{\rho}_i, \hat{\theta}_i)$ which are gained by combined tensile-torsion test lying on the initial yield surface $\Phi(\theta)$. 46

3.4.1 Boundary conditions and dimensions of a quadratic plate under increasing extension $\bar{u}$ on the grey area and uniformly distributed pressure p over the thickness direction. 51

3.4.2 Visualization of yield surface function $\Phi(\theta)$ for (a) $k = 1$ and (b) $k = 0.75$. 52

3.4.3 An illustrative visualization of synthetic tensile data. The first data set (blue, □) is simulated using a single loading path with 10 data points, and the second data set (red, △) consists of two loading paths containing 10 data points. The third data set (black, ∘) consists of 10 loading paths of 10^5 data points. 53

3.4.4 RMSD error of the boundary value problem using the adapted data-driven transition mapping to the reference solution based on the exact material model. The graphs are illustrated concerning the size of tensile data and its corresponding number of loading paths. 54

3.4.5 Contour plot of maximum displacement of the boundary value problem using the adapted data-driven transition mapping. (a, c, e) Maximum displacement and corresponding (b, e, f) absolute errors compared to reference solution at each internal node at time step $t = 150, 300, 450$. The number of loading paths simulating the tensile test is 10 with 10^5 data points. 55

3.4.6 Contour plot of maximum principal stress of the boundary value problem using the adapted data-driven transition mapping. (a, c, e) Maximum principal stress and corresponding (b, e, f) relative errors compared to reference solution at each material point in [Pa] at time step $t = 150, 300, 450$. The number of loading paths simulating the tensile test is 10 with 10^5 data points. 56

4 Article 3: A physics-informed GAN framework based on model-free data-driven computational mechanics

4.3.1 Schematic representation of the neural network's topology, illustrating the progression from input through multiple hidden layers, resulting in displacement $\boldsymbol{u}$ and stress $\boldsymbol{\sigma}$. 64

4.3.2 Schematic representation of the physics-informed neural network's topology, illustrating the progression from input through multiple hidden layers, resulting in displacement $\boldsymbol{u}$ and stress $\boldsymbol{\sigma}$. These outputs undergo automatic differentiation to compute the physics-based loss function $L_{\mathcal{C}}$. The loss is minimized using an optimizer to refine the network parameters θ. 65

4.3.3 Schematic representation of a generative adversarial network (GAN) showcasing the interaction between the generator producing data from random input and the discriminator evaluating the authenticity of both real and generated data. 67

4.3.4 Schematic representation of a physics-informed generative adversarial network (GAN) incorporating collocation points and strain-stress data for physical point generation and discrimination. 69

4.4.1 Illustration of a square plate subjected to external forces, alongside its top-right quadrant representing the symmetry section with specified boundary conditions and force distribution $t(y)$ applied. 71

4.4.2 Visualization of displacement and stress distribution after 200 training epochs offering insights into the material's behavior under the applied loads and conditions (4.35) and (4.36). From top-left to bottom-right: u_x showcases a gradient, indicating a maximum displacement at $(x, y) = (1, 0)$; u_y reveals a displacement trend with negative values highlighted in $x = 0$; σ_x shows a maximum stress magnitude at $x = 0$; σ_y displays a similar gradient at $y = 0$; and σ_{xy} captures a pronounced shear stress distribution inside the plate. 74

4.4.3 Comparative Visualization of the discriminator and generator loss metrics over training iterations for a WGAN-GP model, showcasing the dynamic interplay and convergence patterns. The shaded area shows the maximum range of loss for individual training batches. 74
4.4.4 Visualization of the distance metrics over 200 epochs. The distance of the generated $\boldsymbol{z}$ to the dataset $\mathcal{D}$ illustrates how closely the model-generated outputs match the dataset over training iterations. 75
4.4.5 Comparison of the displacements (u_x, u_y) and stresses $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$ obtained by FEM, WGAN-GP, and PINN along the line $x = y$ within Ω. . . . 76
4.4.6 Visualization of the absolute errors of displacement and stresses in x, y-direction of the WGAN-GP solver compared to the FEM reference solution. 76
4.4.7 Absolute error visualization for displacement and stresses in x, y-directions of the PINN solver, compared to the FEM reference. 77
4.4.8 Global relative error comparison between the WGAN-GP and the standard PINN approach across training epochs. Optimized through hyper-parameter tuning, both models demonstrate convergence towards a lower error boundary. 77

Bibliography

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[2] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, "Machine learning in materials science," *InfoMat*, vol. 1, no. 3, pp. 338–358, 2019.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[4] V. Sundararaghavan and N. Zabaras, "Classification and reconstruction of three-dimensional microstructures using support vector machines," *Computational Materials Science*, vol. 32, no. 2, pp. 223–239, 2005.

[5] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.

[6] F. Nielsen and F. Nielsen, "Hierarchical clustering," *Introduction to HPC with MPI for Data Science*, pp. 195–211, 2016.

[7] R. Cohn and E. Holm, "Unsupervised machine learning via transfer learning and k-means clustering to classify materials image data," *Integrating Materials and Manufacturing Innovation*, vol. 10, no. 2, pp. 231–244, 2021.

[8] A. R. Kitahara and E. A. Holm, "Microstructure cluster analysis with transfer learning and unsupervised learning," *Integrating Materials and Manufacturing Innovation*, vol. 7, pp. 148–156, 2018.

[9] V. Morales-Flórez, M. Piñero, N. De La Rosa-Fox, L. Esquivias, J. A. Anta, and J. Primera, "The cluster model: A hierarchically-ordered assemblage of random-packing spheres for modelling microstructure of porous materials," *Journal of non-crystalline solids*, vol. 354, no. 2-9, pp. 193–198, 2008.

[10] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[11] P. M. Shenai, Z. Xu, and Y. Zhao, "Applications of principal component analysis (pca) in materials science," *Princ. Compon. Anal. Appl*, pp. 25–40, 2012.

[12] A. Zell, *Simulation neuronaler netze*, vol. 1. Addison-Wesley Bonn, 1994.

[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, “Recent advances in convolutional neural networks,” *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[15] S. Dehghan, G. Sattari, S. C. Chelgani, and M. Aliabadi, “Prediction of uniaxial compressive strength and modulus of elasticity for travertine samples using regression and artificial neural networks,” *Mining Science and Technology (China)*, vol. 20, no. 1, pp. 41–46, 2010.

[16] A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi, and L. Song, “Material structure-property linkages using three-dimensional convolutional neural networks,” *Acta Materialia*, vol. 146, pp. 76–84, 2018.

[17] M. B. Gorji, M. Mozaffar, J. N. Heidenreich, J. Cao, and D. Mohr, “On the potential of recurrent neural networks for modeling path dependent plasticity,” *Journal of the Mechanics and Physics of Solids*, vol. 143, p. 103972, 2020.

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[19] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley, “Optimization of molecules via deep reinforcement learning,” *Scientific reports*, vol. 9, no. 1, p. 10752, 2019.

[20] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[22] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, “Recent advances and applications of machine learning in solid-state materials science,” *npj Computational Materials*, vol. 5, no. 1, p. 83, 2019.

[23] Y. Mao, Q. He, and X. Zhao, “Designing complex architectured materials with generative adversarial networks,” *Science advances*, vol. 6, no. 17, p. eaaz4169, 2020.

[24] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[25] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, *et al.*, “Graph neural networks for materials science and chemistry,” *Communications Materials*, vol. 3, no. 1, p. 93, 2022.

[26] L. Torrey and J. Shavlik, “Transfer learning,” in *Handbook of research on machine*

learning applications and trends: algorithms, methods, and techniques, pp. 242–264, IGI global, 2010.

[27] P. I. Frazier and J. Wang, “Bayesian optimization for materials design,” in *Information science for materials discovery and design*, pp. 45–75, Springer, 2015.

[28] H. Yamada, C. Liu, S. Wu, Y. Koyama, S. Ju, J. Shiomi, J. Morikawa, and R. Yoshida, “Predicting materials properties with little data using shotgun transfer learning,” *ACS central science*, vol. 5, no. 10, pp. 1717–1730, 2019.

[29] S. Conti, S. Müller, and M. Ortiz, “Data-driven problems in elasticity,” *Archive for Rational Mechanics and Analysis*, vol. 229, pp. 79–123, Jan 2018.

[30] T. Kirchdoerfer and M. Ortiz, “Data-driven computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 81–101, 2016.

[31] Y. Kanno, “Mixed-integer programming formulation of a data-driven solver in computational elasticity,” *Optimization Letters*, vol. 13, no. 7, pp. 1505–1514, 2019.

[32] M. Röger and B. Schweizer, “Relaxation analysis in a data driven problem with a single outlier,” *Calculus of Variations and Partial Differential Equations*, vol. 59, pp. 1–22, 2020.

[33] T. Kirchdoerfer and M. Ortiz, “Data driven computing with noisy material data sets,” *Computer Methods in Applied Mechanics and Engineering*, vol. 326, pp. 622–641, 2017.

[34] J. A. Sanz-Herrera, J. Mora-Macías, J. Ayensa-Jiménez, E. Reina-Romo, M. H. Doweidar, J. Domínguez, and M. Doblaré, “Data-driven computational simulation in bone mechanics,” *Annals of Biomedical Engineering*, vol. 49, pp. 407–419, 2021.

[35] A. Platzer, A. Leygue, and L. Stainier, “Assessment of data-driven computational mechanics in finite strain elasticity,” in *Constitutive Models for Rubber XI: Proceedings of the 11th European Conference on Constitutive Models for Rubber (ECCMR 2019)*, p. 230, 2019.

[36] A. Platzer, *Finite strain data-driven computational mechanics.: From tailored data to adaptive solvers for multiscale simulations*. PhD thesis, École centrale de Nantes, 2020.

[37] A. Platzer, A. Leygue, L. Stainier, and M. Ortiz, “Finite element solver for data-driven finite strain elasticity,” *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113756, 2021.

[38] L. T. K. Nguyen and M.-A. Keip, “A data-driven approach to nonlinear elasticity,” *Computers & Structures*, vol. 194, pp. 97–115, 2018.

[39] L. T. K. Nguyen, M. Rambausek, and M.-A. Keip, “Variational framework for distance-minimizing method in data-driven computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 112898, 2020.

[40] P. Carrara, L. De Lorenzis, L. Stainier, and M. Ortiz, "Data-driven fracture mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 372, p. 113390, 2020.

[41] P. Carrara, M. Ortiz, and L. De Lorenzis, "Data-driven rate-dependent fracture mechanics," *Journal of the Mechanics and Physics of Solids*, vol. 155, p. 104559, 2021.

[42] T. Kirchdoerfer and M. Ortiz, "Data-driven computing in dynamics," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 11, pp. 1697–1710, 2018.

[43] D. González, F. Chinesta, and E. Cueto, "Thermodynamically consistent data-driven computational mechanics," *Continuum Mechanics and Thermodynamics*, vol. 31, pp. 239–253, 2019.

[44] R. Ibañez, D. Borzacchiello, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladeveze, and F. Chinesta, "Data-driven non-linear elasticity: constitutive manifold construction and problem discretization," *Computational Mechanics*, vol. 60, no. 5, pp. 813–826, 2017.

[45] R. Ibanez, E. Abisset-Chavanne, J. V. Aguado, D. Gonzalez, E. Cueto, and F. Chinesta, "A manifold learning approach to data-driven computational elasticity and inelasticity," *Archives of Computational Methods in Engineering*, vol. 25, no. 1, pp. 47–57, 2018.

[46] D. González, J. V. Aguado, E. Cueto, E. Abisset-Chavanne, and F. Chinesta, "kpca-based parametric solutions within the pgd framework," *Archives of Computational Methods in Engineering*, vol. 25, no. 1, pp. 69–86, 2018.

[47] B. Bahmani and W. Sun, "Manifold embedding data-driven mechanics," *Journal of the Mechanics and Physics of Solids*, vol. 166, p. 104927, 2022.

[48] R. Eggersmann, L. Stainier, M. Ortiz, and S. Reese, "Model-free data-driven computational mechanics enhanced by tensor voting," *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113499, 2021.

[49] R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, and M. Ortiz, "Model-free data-driven inelasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 350, pp. 81–99, 2019.

[50] B. Bahmani and W. Sun, "A kd-tree-accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data," *Computer Methods in Applied Mechanics and Engineering*, vol. 382, p. 113868, 2021.

[51] H. Salahshoor and M. Ortiz, "Model-free data-driven viscoelasticity in the frequency domain," *Computer Methods in Applied Mechanics and Engineering*, vol. 403, p. 115657, 2023.

[52] T. Bartel, M. Harnisch, B. Schweizer, and A. Menzel, "A data-driven approach for plasticity using history surrogates: Theory and application in the context of truss

structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 414, p. 116138, 2023.

[53] K. Poelstra, T. Bartel, and B. Schweizer, "A data-driven framework for evolutionary problems in solid mechanics," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 103, no. 3, p. e202100538, 2023.

[54] J. Yang, W. Huang, Q. Huang, and H. Hu, "An investigation on the coupling of data-driven computing and model-driven computing," *Computer Methods in Applied Mechanics and Engineering*, vol. 393, p. 114798, 2022.

[55] E. Marenic, G. Seychal, and J.-C. Passieux, "Data driven approach in multiphysics framework: Application to coupled electro-mechanical problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 395, p. 114959, 2022.

[56] K. Karapiperis, M. Ortiz, and J. Andrade, "Data-driven nonlocal mechanics: Discovering the internal length scales of materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 386, p. 114039, 2021.

[57] R. Xu, J. Yang, W. Yan, Q. Huang, G. Giunta, S. Belouettar, H. Zahrouni, T. B. Zineb, and H. Hu, "Data-driven multiscale finite element method: From concurrence to separation," *Computer Methods in Applied Mechanics and Engineering*, vol. 363, p. 112893, 2020.

[58] K. Karapiperis, L. Stainier, M. Ortiz, and J. Andrade, "Data-driven multiscale modeling in mechanics," *Journal of the Mechanics and Physics of Solids*, vol. 147, p. 104239, 2021.

[59] J. Mora-Macías, J. Ayensa-Jiménez, E. Reina-Romo, M. H. Doweidar, J. Domínguez, M. Doblaré, and J. A. Sanz-Herrera, "A multiscale data-driven approach for bone tissue biomechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 368, p. 113136, 2020.

[60] X. Guo, Z. Du, C. Liu, and S. Tang, "A new uncertainty analysis-based framework for data-driven computational mechanics," *Journal of Applied Mechanics*, vol. 88, no. 11, p. 111003, 2021.

[61] S. Zschocke, F. Leichsenring, W. Graf, and M. Kaliske, "A concept for data-driven computational mechanics in the presence of polymorphic uncertain properties," *Engineering Structures*, vol. 267, p. 114672, 2022.

[62] T. F. Korzeniowski, *Data-driven finite element computation with material uncertainty*. Ph.d. dissertation, Universität Siegen, Siegen, Germany, 2022.

[63] A. Leygue, M. Coret, J. Réthoré, L. Stainier, and E. Verron, "Data-based derivation of material response," *Computer Methods in Applied Mechanics and Engineering*, vol. 331, pp. 184–196, 2018.

[64] A. Leygue, R. Seghir, J. Réthoré, M. Coret, E. Verron, and L. Stainier, "Non-

parametric material state field extraction from full field measurements," *Computational Mechanics*, vol. 64, no. 2, pp. 501–509, 2019.

[65] L. Stainier, A. Leygue, and M. Ortiz, "Model-free data-driven methods in mechanics: material data identification and solvers," *Computational Mechanics*, pp. 1–13, 2019.

[66] M. Dalémat, M. Coret, A. Leygue, and E. Verron, "Measuring stress field without constitutive equation," *Mechanics of Materials*, vol. 136, p. 103087, 2019.

[67] M. Dalémat, M. Coret, A. Leygue, and E. Verron, "Reliability of the data-driven identification algorithm with respect to incomplete input data," *Constitutive Models for Rubber XI*, pp. 311–316, 2019.

[68] T.-H. Su, J. G. Jean, and C.-S. Chen, "Model-free data-driven identification algorithm enhanced by local manifold learning," *Computational Mechanics*, vol. 71, no. 4, pp. 637–655, 2023.

[69] J. Ayensa-Jiménez, M. H. Doweidar, J. A. Sanz-Herrera, and M. Doblaré, "A new reliability-based data-driven approach for noisy experimental data with physical constraints," *Computer Methods in Applied Mechanics and Engineering*, vol. 328, pp. 752–774, 2018.

[70] T. F. Korzeniowski and K. Weinberg, "A comparison of stochastic and data-driven fem approaches to problems with insufficient material data," *Computer Methods in Applied Mechanics and Engineering*, vol. 350, pp. 554–570, 2019.

[71] E. Prume, S. Reese, and M. Ortiz, "Model-free data-driven inference in computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 403, p. 115704, 2023.

[72] S. Conti, F. Hoffmann, and M. Ortiz, "Model-free and prior-free data-driven inference in mechanics," *Archive for Rational Mechanics and Analysis*, vol. 247, no. 1, p. 7, 2023.

[73] R. Eggersmann, L. Stainier, M. Ortiz, and S. Reese, "Efficient data structures for model-free data-driven computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 382, p. 113855, 2021.

[74] B. van der Heijden, Y. Wang, and G. Lubineau, "Frankenstein's data-driven computing approach to model-free mechanics," *Computational Mechanics*, vol. 71, no. 6, pp. 1269–1280, 2023.

[75] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[76] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[77] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in neural information processing systems*, vol. 15, 2002.

[78] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[79] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[80] K. Ciftci and K. Hackl, “Model-free data-driven inelasticity in haigh-westergaard space – a study how to obtain data points from measurements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 416, p. 116352, 2023.

[81] J. Ghaboussi, J. H. Garrett, and X. Wu, “Knowledge-based modeling of material behavior with neural networks,” *Journal of Engineering Mechanics-asce*, vol. 117, pp. 132–153, 1991.

[82] B. Hkdh, “Neural networks in materials science,” *ISIJ international*, vol. 39, no. 10, pp. 966–979, 1999.

[83] W. Sha and K. Edwards, “The use of artificial neural networks in materials science based research,” *Materials & Design*, vol. 28, no. 6, pp. 1747–1752, 2007.

[84] A. Zhang and D. Mohr, “Using neural networks to represent von mises plasticity with isotropic hardening,” *International Journal of Plasticity*, vol. 132, p. 102732, 2020.

[85] I.-C. Yeh, “Modeling of strength of high-performance concrete using artificial neural networks,” *Cement and Concrete research*, vol. 28, no. 12, pp. 1797–1808, 1998.

[86] J. F. Unger and C. Könke, “Neural networks as material models within a multiscale approach,” *Computers & structures*, vol. 87, no. 19-20, pp. 1177–1186, 2009.

[87] C. Huang, L. Zhang, L. He, J. Sun, B. Fang, B. Zou, Z. Li, and X. Ai, “A study on the prediction of the mechanical properties of a ceramic tool based on an artificial neural network,” *Journal of Materials Processing Technology*, vol. 129, no. 1-3, pp. 399–402, 2002.

[88] Y. Yousif, K. Daws, and B. Kazem, “Prediction of friction stir welding characteristic using neural network,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 2, no. 3, pp. 151–155, 2008.

[89] S. Feng, H. Zhou, and H. Dong, “Using deep neural network with small dataset to predict material defects,” *Materials & Design*, vol. 162, pp. 300–310, 2019.

[90] K. Wang and W. Sun, “A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning,” *Computer Methods in Applied Mechanics and Engineering*, vol. 334, pp. 337–380, 2018.

[91] R. E. Jones, J. A. Templeton, C. M. Sanders, and J. T. Ostien, “Machine learning mod-

els of plastic flow based on representation theory," *arXiv preprint arXiv:1809.00267*, 2018.

[92] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa, "Deep learning predicts path-dependent plasticity," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, pp. 26414–26420, 2019.

[93] N. N. Vlassis, R. Ma, and W. Sun, "Geometric deep learning for computational mechanics part i: Anisotropic hyperelasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 371, p. 113299, 2020.

[94] A. Koujelev, M. Sabsabi, V. Motto-Ros, S. Laville, and S. Lui, "Laser-induced breakdown spectroscopy with artificial neural network processing for material identification," *Planetary and Space Science*, vol. 58, no. 4, pp. 682–690, 2010.

[95] J. G. Greener, L. Moffat, and D. T. Jones, "Design of metalloproteins and novel protein folds using variational autoencoders," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[96] A. Galetzka, D. Loukrezis, and H. D. Gersem, "Data-driven solvers for strongly nonlinear material response," *ArXiv*, vol. abs/2008.08482, 2020.

[97] P. Mordohai and G. Medioni, "Dimensionality estimation, manifold learning and function approximation using tensor voting," *Journal of Machine Learning Research*, vol. 11, pp. 411–450, 01 2010.

[98] S. Timoshenko, *History of strength of materials: with a brief account of the history of theory of elasticity and theory of structures.* Courier Corporation, 1983.

[99] A. Galetzka, D. Loukrezis, and H. De Gersem, "Data-driven solvers for strongly nonlinear material response," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 6, pp. 1538–1562, 2021.

[100] A. Galetzka, D. Loukrezis, and H. De Gersem, "Three-dimensional data-driven magnetostatic field computation using real-world measurement data," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 41, no. 2, pp. 615–627, 2022.

[101] K. Ciftci and K. Hackl, "Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules," *Computational Mechanics*, vol. 70, pp. 425–435, May 2022.

[102] P. Menétrey and K. J. Willam, "Triaxial failure criterion for concrete and its generalization," *Aci Structural Journal*, vol. 92, pp. 311–318, 1995.

[103] H. Jiang and Y. Yang, "A three-dimensional hoek–brown failure criterion based on an elliptical lode dependence," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 44, no. 18, pp. 2395–2411, 2020.

[104] P. S. nski, "Yield criterion accounting for the influence of the third invariant of stress

tensor deviator. part ii. analysis of convexity condition of the yield surface," *Engineering Transactions*, vol. 59, no. 4, 2014.

[105] C. Kim, H. Son, and C. Kim, "Fully automated registration of 3d data to a 3d cad model for project progress monitoring," *Automation in Construction*, vol. 35, pp. 587–594, 2013.

[106] Q. Peng and M. Chen, "An efficient return mapping algorithm for general isotropic elastoplasticity in principal space," *Computers & Structures*, vol. 92-93, pp. 173–184, 2012.

[107] L. T. K. Nguyen, R. C. Aydin, and C. J. Cyron, "Accelerating the distance-minimizing method for data-driven elasticity with adaptive hyperparameters," *Computational Mechanics*, vol. 70, no. 3, pp. 621–638, 2022.

[108] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7, 2001.

[109] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," *Advances in neural information processing systems*, vol. 30, 2017.

[110] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[111] J. D. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning*, 2001.

[112] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[113] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020.

[114] L. Sun, H. Gao, S. Pan, and J.-X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Computer Methods in Applied Mechanics and Engineering*, vol. 361, p. 112732, 2020.

[115] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

[116] A. Dourado and F. A. Viana, "Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 6, 2020.

[117] J.-H. Bastek and D. M. Kochmann, "Physics-informed neural networks for shell structures," *European Journal of Mechanics-A/Solids*, vol. 97, p. 104849, 2023.

[118] Q. He, D. Barajas-Solano, G. Tartakovsky, and A. M. Tartakovsky, "Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport," *Advances in Water Resources*, vol. 141, p. 103610, 2020.

[119] M. Yin, X. Zheng, J. D. Humphrey, and G. E. Karniadakis, "Non-invasive inference of thrombus material properties with physics-informed neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 375, p. 113603, 2021.

[120] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE transactions on neural networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[121] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, 2021.

[122] L. Yang, X. Meng, and G. E. Karniadakis, "B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data," *Journal of Computational Physics*, vol. 425, p. 109913, 2021.

[123] T. Kadeethum, T. M. Jørgensen, and H. M. Nick, "Physics-informed neural networks for solving inverse problems of nonlinear biot's equations: Batch training," in *54th US Rock Mechanics/Geomechanics Symposium*, OnePetro, 2020.

[124] J. Platt and A. Barr, "Constrained differential optimization," in *Neural Information Processing Systems*, 1987.

[125] C.-L. Hwang and A. S. M. Masud, *Multiple objective decision making-methods and applications: a state-of-the-art survey*, vol. 164. Springer Science & Business Media, 2012.

[126] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3055–A3081, 2021.

[127] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *Journal of Computational Physics*, vol. 404, p. 109136, 2020.

[128] A. D. Jagtap, K. Kawaguchi, and G. Em Karniadakis, "Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks," *Proceedings of the Royal Society A*, vol. 476, no. 2239, p. 20200334, 2020.

[129] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations," *Journal of Computational Physics*, vol. 426, p. 109951, 2021.

[130] A. Henkes, H. Wessels, and R. Mahnken, “Physics informed neural networks for continuum micromechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 393, p. 114790, 2022.

[131] Y. Yang and P. Perdikaris, “Adversarial uncertainty quantification in physics-informed neural networks,” *Journal of Computational Physics*, vol. 394, pp. 136–152, 2019.

[132] A. Daw, M. Maruf, and A. Karpatne, “Pid-gan: A gan framework based on a physics-informed discriminator for uncertainty quantification with physics,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 237–247, 2021.

[133] B. Bullwinkel, D. Randle, P. Protopapas, and D. Sondak, “DEQGAN: Learning the loss function for PINNs with generative adversarial networks,” 2022.

[134] L. T. K. Nguyen, M. Rambausek, and M.-A. Keip, “Variational framework for distance-minimizing method in data-driven computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 112898, 2020.

[135] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[136] J. Berg and K. Nyström, “A unified deep artificial neural network approach to partial differential equations in complex geometries,” *Neurocomputing*, vol. 317, pp. 28–41, 2018.

[137] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, “Physics-informed neural networks (pinns) for fluid mechanics: A review,” *Acta Mechanica Sinica*, pp. 1–12, 2022.

[138] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[139] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for gans do actually converge?,” in *International conference on machine learning*, pp. 3481–3490, PMLR, 2018.

[140] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” *Advances in neural information processing systems*, vol. 30, 2017.

[141] I. M. Sobol’, “On the distribution of points in a cube and the approximate evaluation of integrals,” *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[142] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *Advances in neural information processing systems*, vol. 29, 2016.

[143] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.

[144] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Kerem Ciftci

PERSONAL DETAILS

Curriculum Vitae

Day of Birth	June 3rd, 1990
Place of Birth	Siegen, Germany
Nationality	German
Residence	Cologne, Germany

ACADEMIC CAREER

Ph.D. Student since 2019

Institute of Mechanics of Materials, Dept. of Civil and Environmental Engineering, Ruhr-University Bochum, Germany

Research Visiting Student 2019

Vietnamese-German University, Ho Chi Minh City, Vietnam

M. Sc. Computational Engineering 2017-2019

Ruhr-University Bochum, Germany

Study Abroad 2016

EWHA University, Seoul, South-Korea

B.Sc. Mathematics 2011-2016

University of Cologne, Germany

Higher School Certificate 2008-2010

Hölderlin Gymnasium, Cologne, Germany

PROFESSIONAL CAREER

Research Associate since 2019

Institute of Mechanics of Materials, Dept. of Civil and Environmental Engineering, Ruhr-University Bochum, Germany

Civilian Service, Volunteer Tutoring at Youth Center since 2023

Cologne branches, Germany

Intern 2016

Bayer AG, Leverkusen, Germany

Graduate Assistant in Financial Mathematics 2015

Technische Hochschule Köln, Cologne, Germany

Student Assistant in Differential Equations and Numerics 2013 - 2015

University of Cologne, Germany

Community service 2010-2011

AWO Seniorenzentrum Saaler Mühle, Bergisch Gladbach, Germany

Mitteilungen aus dem Institut für Mechanik
RUHR-UNIVERSITÄT BOCHUM
Nr. 195

ISBN 978-3-935892-73-5