Data-driven finite element computation with material uncertainty

DISSERTATION
zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften

vorgelegt von
M.Sc. Tim Fabian Korzeniowski

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2022

Betreuerin und erste Gutachterin
Prof. Dr.-Ing. Kerstin Weinberg
Universität Siegen

Zweite Gutachterin
Prof. Dr.-Ing. habil. Stefanie Reese
Universität Aachen

Tag der mündlichen Prüfung
27.06.2022

Abstract

Data-driven methodologies are attracting more and more attention. The "data-rich" world ushered in the 21st century with ever larger and more unwieldy data sets directed interest toward processing them. Gradually, data-driven computations are also gaining ground in the natural sciences. In numerical mechanics, such methodologies are used in particular for describing the material behavior. Normally, the behavior has to be characterized via experiments and then fitted into models. The second step can be omitted if one is able to use the data obtained by the experiments directly in the calculation. This methodology was presented in 2016 by Trenton Kirchdoerfer and Michael Ortiz for a finite element analysis and will be followed up in this work.

First, the so-called "data-driven finite element method" is investigated with respect to its different input parameters. It turns out that the method is robust concerning the numerical stiffness, but the amount and quality of data is essential for the quality of the solution. Then this methodology is investigated in terms of material uncertainties. For this purpose, data-driven computations are performed with data that are generated from stochastic fields. In order to limit the additional numerical effort of the data-driven computation, a method is presented, which by means of a multi-level computation is able to reduce the numerical effort. Instead of a simulation with the complete data set, several simulations with small, adaptive data sets are used. To avoid the complex experimental generation of three-dimensional data, a methodology is presented on how data from numerical calculations can be obtained. Here using the example of a representative volume element. Finally, two further application examples are presented: First, the data-driven methodology is applied to a diffusion problem instead of a mechanical problem. Second, a polymorphic uncertainty model is generated by adding a fuzzy variable as a further uncertainty to the model.

keywords: data-driven, finite element method, stochastic finite element method, uncertainties, multi-level model, representative volume element, diffusion, polymorphic uncertainty.

Zusammenfassung

Datengetriebene Methodiken ziehen immer mehr Aufmerksamkeit auf sich. Die im 21. Jahrhundert eingeläutete „datenreiche“ Welt mit immer größeren und unübersichtlicheren Datenmengen lenkt das Interesse auf die Verarbeitung der Datensätze. Nach und nach erhalten datengetriebene Rechnungen auch in den Naturwissenschaften Einzug. In der numerischen Mechanik werden solche Methodiken insbesondere für das Beschreiben des Materialverhaltens genutzt. Normalerweise muss das Verhalten über Experimente charakterisiert und anschließend an Modelle angepasst werden. Auf den zweiten Schritt kann verzichtet werden, wenn man in der Lage ist, die aus den Experimenten erhaltenen Daten direkt in der Rechnung zu verwenden. Diese Methodik wurde 2016 von Trenton Kirchdoerfer und Michael Ortiz für die Finite-Elemente-Methode vorgestellt und soll in dieser Arbeit weiterverfolgt werden.

Zunächst wird die sogenannte „datengetriebene Finite-Elemente-Methode“ hinsichtlich ihrer unterschiedlichen Inputparameter untersucht. Es stellt sich heraus, dass das Verfahren sehr robust gegenüber Schwankungen der numerischen Steifigkeit ist, die Datenmenge aber essentiell für die Qualität der Lösung ist. Mit Blick auf die Datenmenge wird dann untersucht, wie geeignet diese Methodik dazu ist, Materialunsicherheiten zu implementieren. Dazu werden die datengetriebenen Rechnungen mit Daten durchgeführt, die aus stochastischen Feldern generiert werden. Um den numerischen Mehraufwand der datengetriebenen Rechnung zu begrenzen wird ein Verfahren vorgestellt, dass mittels einer mehrstufigen Rechnung in der Lage ist, den numerischen Aufwand zu verringern. Anstatt einer Simulation mit der kompletten Datenmenge, wird auf mehrere Simulationen mit kleinen, adaptiven Datenmengen gesetzt. Um die komplexe experimentelle Generierung drei-dimensionaler Daten zu umgehen wird eine Methodik vorgestellt, wie Daten aus numerischen Rechnungen, hier am Beispiel eines repräsentativen Volumenelements, gewonnen werden können. Zuletzt werden zwei weitere Anwendungsbeispiele vorgestellt: Zum Einen wird die datengetriebene Methodik auf das Diffusionsproblem angewendet anstatt auf ein mechanisches Problem. Zum Anderen wird ein polymorphes Unschärfemodell generiert indem eine Fuzzy-Variable als weitere Unsicherheit in das Modell hinzugenommen wird.

Schlagwörter: datengetrieben, Finite-Elemente-Methode, stochastische Finite-Elemente-Methode, Unsicherheiten, Mehrstufenmodell, repräsentatives Volumenelement, Diffusion, polymorphe Unschärfe

Acknowledgments

First of all, I would like to thank everyone who helped me on the way to writing this thesis. My biggest thanks go to Prof. Dr. Weinberg, who gave me the confidence to work on the DFG project at her chair. I would like to thank her for all the wisdom, help, knowledge and support she shared with me. Furthermore I want to thank Prof. Dr. Stefanie Reese for taking over the co-referee, Prof. Dr. Oliver Nelles and Prof. Dr. Peter Kraemer for joining and completing the thesis committee.

Further on, I want to thank all my fellow groupmates for all the discussions, tips and advice they gave to me. My sincere thanks go to Carola Bilgen, who especially helped me in the beginning of my time at the Chair of Solid Mechanics with many fruitful discussions and advice. A special thanks also goes to Ines Münker, who always helped me keep a smile on my face during all tasks I had to solve in the last years.

Contents

Notations and conventions

If ever the overview within the three different finite element methods and their variables is lost, have a look here. An incomplete list of symbols is given here which may look similar but have different meanings:

FEM	finite element method
SFEM	stochastic finite element method
DD-FEM	data-driven finite element method
RVE	representative volume element
UQ	uncertainty quantification
Ω	body of the considered object
Ω_e	one finite element of the body
Γ	boundary of Ω
Γ_u	Dirichlet boundary
Γ_σ	Neumann boundary
λ	1. Lamé parameter
$\boldsymbol{\lambda}$	Lagrange multiplier for the DD-FEM
$\mathbb{C}$	stiffness tensor
$\mathbb{C}^\mu$	mean stiffness tensor (SFEM)
$\mathbb{C}^\circ$	num. stiffness tensor (DD-FEM)
$\mathbb{D}$	compliance tensor
$\mathbb{D}^\circ$	num. compliance tensor (DD-FEM)
$\mathbb{D}_1$	diffusion tensor (Section 7.2)
$\mathcal{D}$	data set containing the constitutive data points
$\mathcal{D}(n)$	data set containing n constitutive data points per index
$\Pi(\mathbf{u})$	total potential energy
$\Psi^{\mathrm{el}}(\epsilon)$	elastic energy density (classic)
$\Psi^{\mathrm{el}}_\chi(\epsilon)$	elastic energy density (SFEM)
$\Psi^{\mathrm{el}}_\circ(\epsilon)$	elastic energy density (DD)
$\Psi^*(\sigma)$	conjugate energy density (classic)
$\Psi^*_\circ(\sigma)$	conjugate energy density (DD-FEM)
$\Psi_\circ(\epsilon, \sigma)$	local minimization functional (DD-FEM)
W^{py}	global penalty functional
W^*	mixed energy functional
$e = 1, ..., n_e$	numbering of the elements
$m = 1, ..., n_m$	numbering of the integration points

$\mathcal{N}_{(a,b^2)}$	normal distribution with mean a and variance b^2
$N_{i,j}$	derivative of the i-th shape function with respect to j
E	Young's modulus
$\mathbb{E}(\bullet)$	expected value of random variable $\bullet$

1 Introduction

Many data science methods have come to the forefront in the last twenty years. The ability to save, order and process enormous volumes of data made data-driven methods easier to handle and very attractive. Turing Award winner Jim Gray predicted a 'fourth paradigm' of science, after the empirical, theoretical and computational one, and stated: "everything about science is changing because of the impact of information technology and the data deluge" [49]. A real hype around the buzzword "data science" emerged [1, 2, 87]. Even many non-scientific people are in contact with data science today since the rise of Facebook, Amazon and Google as global players and their individual-related data desire. From social sciences, healthcare, e-commerce to mobile communications, it influences our daily life significantly [52]. However, in the natural sciences, data science methods only increased slowly and the full potential has not been reached yet. The underlying concept of data methods can generally be described in the three steps collect, analyze and interpret as follows:

1. Collect as much data as possible about a certain process, item or phenomenon, for instance:
   i) manufacturing process of a component C
   ii) water mark level W
   iii) purchase of a person P
2. Use analysis tools (often statistical analysis tools), e.g. principal component analysis, neural networks, market basket analysis, sensitivity analysis
3. Gain knowledge about relations and correlations, which may be unknown before and predict further outcomes. Building on the prior examples:
   i) component C fails the fatigue test if the temperature during the manufacturing process is too low
   ii) risk of a flood is high in the future if water mark W exceeds certain levels
   iii) after person P bought a product send him advertising to affiliated products

The fields of application are not limited. However, critics also arose if data science is something different than statistics [113, 114].

The term data science was used freely from the 1960s until the '90s for computational and statistic methods [1, 87]. In 1992, in a statistics symposium in Montpellier, it was

recognized that there is a special need for a new discipline regarding the use of data of different dimensions, origins, and structures. Since then, even though missing a clear formal definition until today, data science has been connected to the extraction of knowledge from data. And even though Usama Fayyad et al. wrote in 1996: "Historically, the notion of finding useful patterns in data has been given a variety of names, including data mining, knowledge extraction, information discovery, information harvesting, data archeology, and data pattern processing...", data science established itself as the key term for knowledge extraction from data [38].

Figure 1.1: Various terms which are used in conjunction with the buzzword data science, from [4].

Data science in mechanical engineering

Data science applications are still used more in commercial fields than in the STEM (Science, Technology, Engineering and Mathematics) fields until now [49]. In engineering fields, data science applications begin to rise slowly. A significant difference is that in the

engineering fields a general relationship is mostly known and a quantification is missing - it is usually not about finding useful patterns or unknown relationships between several features. Furthermore, the data problems are often small in contrast to the big data problems of other fields. In engineering experiments, big data can also be produced in high-energy physics, but Petabytes that can be produced per day are rather seldom [52]. In addition to that, the most essential difference is that commercial applications are comparatively simple in contrast to scientific problems.

In computational mechanics, especially machine learning approaches are investigated for describing the material behavior. Instead of a constitutive equation, a constitutive relationship is gained by neural networks for example. In recent works [55, 56, 102] data-driven methods are employed for fatigue for example. Other researchers use this approach in the framework of inelasticity, plasticity, hyperelasticity and fluid flow [52, 53, 46, 76].

Data-driven computational mechanics

This thesis is based on the work of Trenton Kirchdoerfer and Michael Ortiz [67] who first proposed a data-driven finite element method (DD-FEM) in 2016. This new concept is characterized by the usage of experimental data directly in a finite element simulation. Measured strain-stress data from experiments are gathered in a data set and used as an input for the finite element analysis to describe the material behavior. The main advantage is that the material behavior can be described model-free, i.e. no constitutive equation needs to be chosen and the empirical modeling step is elided, which overcomes errors and uncertainties in the material modeling step. The growing use of novel and

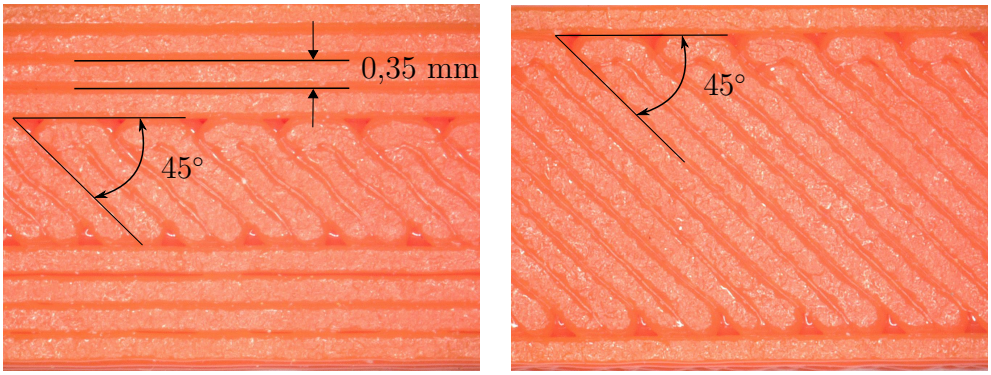


Figure 1.2: Two examples of additive manufactured tension test specimen. While they have the same matrix material with known parameters they still behave different. Data-driven approaches may help in this example [95].

specifically designed materials, such as compounds or the use of additive manufacturing,

comes along with the challenge of their identification and modeling, see Figure 1.2. Both components use the same matrix material, however, the different printing patterns lead to the different behavior of both components.

Often the basic properties of such materials are not well known, making it difficult to extract constitutive parameters reliably. Besides the general material behavior, the material's fluctuations are also of interest. Generally, the experiments which determine the material behavior are repeated with multiple samples. Therefore, the natural material fluctuations caused by imperfections, impurities, gas inclusions or the production process are also included in the data sets that serve as material input of the data-driven finite element method.

Material uncertainty

Within the priority project 1886 'Polymorphic Uncertainty Modeling for the Numerical Design of Structures' of the German Research Association (DFG) it is the task to focus on methods that are able to describe uncertainties and to use them in polymorphic approaches. In a deterministic finite element analysis, we restrict ourselves to averaged values of stiffnesses or loads that attack an idealized geometry. The common practice in engineering to overcome those uncertainties are safety factors. However, that means we are unable to quantify or predict the influence of sources of randomness. The consideration of material uncertainty or other uncertainties (geometric, boundary conditions) in general and their evaluation falls under the category of uncertainty quantification.

The project's first attempt was the use of a stochastic Markov process [70]. However, this turned out to be unsatisfactory, so that the data-driven methodology mentioned above was investigated with regard to its usability. Today, the main approach to treat uncertainties is often a stochastic one. However, stochastic frameworks are not always used. Especially if there is only small and limited knowledge available, it is hard to justify assumptions on the stochastic distributions and fuzzy approaches are used as well. Obviously, the material data set can only be used to investigate material uncertainties. In this regard, it is researched if the data-driven approach can describe the material variations using only the data set, similarly to the material law leading to a model-free uncertainty approach. Alongside stochastic and fuzzy methods data-driven methodologies could then take on a new class of uncertainty methods .

In this thesis, material uncertainties are considered as the data which is used as an input to the finite element simulation automatically includes the material fluctuations. That means material uncertainties are included without using a general model for them. Therefore, we will compare the data-driven approach to the common approach to include material uncertainties into finite element computations, namely the stochastic finite element method.

1.1 State of the art and objectives of this thesis

Data science methods are introduced more and more in the STEM fields. In mechanical engineering, they especially get employed for the constitutive behavior of materials. Several data-driven methods were proposed there in the last years [18, 90, 98, 100]. In particular neural networks are applied to model the material behavior and to elide a constitutive equation [40, 48, 50, 83]. Those constitutive models are then used in finite element analyses. Besides mechanical problems, which are considered in this thesis, it can be used for heat transfer, mass transfer, electromagnetic and other boundary value problems. Today, the finite element method is a widely accepted method for solving partial differential equations in engineering fields. Those problems can mainly be characterized by the three equations depicted in Fig. 1.3

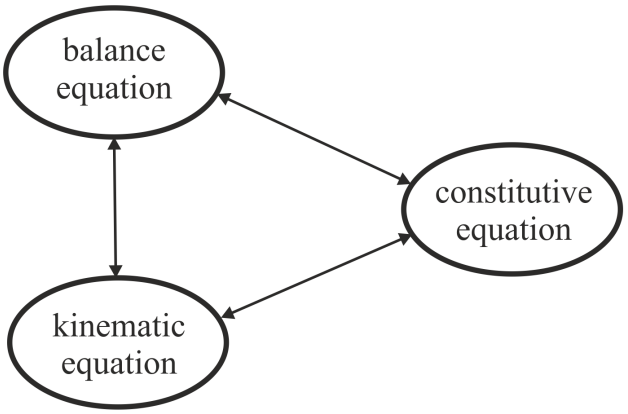


Figure 1.3: The three equations which fundamentally determine a mechanical field problem.

In the mechanical case, the balance equation puts outer and inner forces which work onto the body, in balance. The kinematic equation describes the relationship between displacements and strains, and the constitutive equation describes the material behavior, i.e. the strain-stress relationship. To describe the material in a finite element analysis, a constitutive relationship needs to be utilized. Typically, a functional relationship in the sense of a strain energy density is given, which describes the relationship between the stresses and the strains. In 2016 Trenton Kirchdoerfer and Michael Ortiz introduced the data-driven finite element method. In this algorithm, they propose to use measured strain-stress data directly in the finite element method to describe the material's behavior instead of using a constitutive equation. If the empirical fit that normally follows the experiments can be omitted, the model uncertainties can be reduced entirely. Furthermore, the data-driven method allows us to use the information gained during the experiment as pure as possible. Using the measured data itself as an input into a numerical simulation, we use every information gained during the experiment, and it is no

further material behavior predicted. There is no need to describe the shape of data by a mathematical function, nor the fluctuations of the data need to be described. They go into the simulation as they are. Those promising attributes of the data-driven finite element method make it worth investigating this new method.

As a new approach, Kirchdoerfer and Ortiz introduced the data-driven algorithm for truss structures and linear elasticity [67]. As the discrete data set and the balance laws will have no intersection in general, we aim for minimizing the distance between the data set and the balance laws. A distance minimizing problem emerges then. In between a distance in the phase space needs to be defined that measures strain and stresses adequately. Energy densities adopted from linear elasticity are employed to determine the distance in the phase space which, however, do not restrict the method to linear material behaviors. In their pivotal publication, they present good convergence properties of their proposed algorithm towards the number of data points and the assignment of data points as the solution. The robustness towards the spatial discretization is also outlined. In the publication, great emphasis is placed on the mathematical basis of the methodology, but also pose questions: How many data points are needed for an acceptable result? How well does the method behave with the typical scattering of experimental data? What prior treatments can be applied to the data?

They continued their work towards the data-driven finite element method with two additional publications. In [68] the former distance-minimizing data-driven algorithm was generalized to the so-called max-entropy data-driven algorithm. Cluster analysis was used to make the algorithm robust towards outliers in the data set. The distance-minimizing algorithm is recovered for the case of infinite temperature. Lastly, in their publication [69] they extend their data-driven algorithm from static to the dynamic framework, thus including time integration and the possibility of time-dependent data sets.

Since then, a handful of publications have dealt with the newly proposed data-driven algorithm by Kirchdoerfer and Ortiz. The works of Conti et al. [25, 26, 27] focus on the mathematical framework of the data-driven problem. They identify the conditions for the convergence and the well-posedness of the data-driven problem in the elastic and finite elastic setting. Further publications try to extend the data-driven method to non-linear and time-dependent problems. In [88], Nguyen and Keip extend the DD-FEM to finite strain theory, while they present a variational framework of the data-driven problem for a classical Poisson equation in [89], which is detached from the mechanical problem. Eggersmann et al. extend the DD-FEM to history-dependent inelastic material data in their first publication [34]. Their second publication [36] extends the measured data set by pointwise tangent spaces to give the data set an underlying structure. Finally, they focus on the efficient computation of the nearest data points in [35] by approximate nearest-neighbor methods.

The group of Stainier and Leygue also worked extensively on the data-driven framework and how to derive data sets. In the works [29, 79, 80, 97, 103] they focus on the

derivation of the data set from image analysis. Together with Platzer, they focus on finite strain elasticity in [93] and [94]. Further publications with Dalémat focus on the robustness and reliability of the data-driven method towards incomplete data [30, 31]. The publications of Ibanez and Chinesta focus on the construction of a constitutive manifold [52, 54] and plasticity [22, 53]. Kanno focuses on the usage of statistical methods on the material data set. In [60] he uses a k-means clustering method to extend the data-driven algorithm more robust towards noise and outliers. In [59, 61, 62] he uses a kernel regression ansatz for coarse data sets to gain unknown stress values for given strain values. Further publications tackle the problem of fracture mechanics [20] and the usage of lower dimensional data [106, 107, 108].

Even though the algorithm has received increasing attention over the last five years there are still plenty of questions to answer. The questions tackled in this thesis are posed from a more pragmatic point of view. The research presented is developed inside the Priority Program 1886 focusing on polymorphic uncertainty modeling. Therefore, a main emphasis of this thesis will be the ability to include the material uncertainty inside the data set. A comparison with existing methods is made and compared to the ability of the data-driven method to include uncertain material behavior. While the works of Kirchdoerfer and Ortiz set the theoretical basis of the data-driven finite element method, a deep and broad analysis of the working method is missing. How does the data influence the solution? How does the numerical stiffness influence the solution? How does the random initial assignment influence the solution? All these are questions that will be answered in the course of this work. Also, in most of the previously mentioned publications, three-dimensional problems are avoided due to the numerical cost of the data-driven method. A look at this problem is laid out in a separate chapter. Lastly, we tackle the question of how to gain a data set. It can be rather complicated and expensive to gain experimental data due to the dimension of the data. Therefore, in most of the publications a synthetically generated data set is used. One chapter is devoted to the meaningful generation of a numerical data set.

1.2 Outline of this thesis

For this thesis, the data-driven framework was implemented into an existing in-house finite element code in MATLAB [5]. All data-driven simulations and finite element computations with a functional constitutive equation[1], which are conducted for comparison reasons, are run on this code. Computations of representative volume elements are conducted in ABAQUS [6]. In the following, the structure is clarified briefly by chapters:
Chapter 2 “Governing equations” lies the fundamentals of the different finite element methods. We repeat the basic and fundamental equations of the linear finite element

[1]The finite element computations with a functional relationship will be called classical FEM.

method. Built upon this basis, the extension to the stochastic finite element method is given before the equation systems of the data-driven finite element method are derived. We conclude the chapter with a comparison of the equation systems that need to be solved in each case.

Chapter 3 "Numerical realization" concentrates on the numerical implementation of the finite element methods. In a brief general section we focus on numerical aspects, such as the shape function, numerical integration and numerical implementation. In a section about the stochastic finite element method, we focus on the realization of the random field and the Monte Carlo method, which is used to derive the statistics of the random solution. The data-driven section focuses on the synthetic generation of data sets as the set is the crucial description of the material behavior in the data-driven computation. Before we end the chapter with the different numerical problems used in the thesis, the fixed-point algorithm of the DD-FEM is introduced.

In Chapter 4 "Data-driven, classical and stochastic FEM" first results of this thesis are displayed. We start with the basic properties of the data-driven method shown in a simple rod example. Moving on to more dimensions, we start comparing the different methods. Depending on the two input variables of the data-driven method - the data set and the numerical stiffness tensor - we compare the results to the classical deterministic finite element method. Afterwards, we investigate the ability of the data-driven method to include material uncertainties within the data set and compare it to the stochastic finite element method.

Chapter 5 "Multi-level data set approach" focuses on a numerical improvement of the data-driven finite element method. The search for the closest data points, which is executed in every data iteration step, is numerical very costly. We address this problem here by using adaptive data sets. We start with a coarse set to identify those areas of the strain-stress phase space in which precise data is needed. Then we increase precision by successive addition of data points in these areas.

The data needed for the data-driven method does not need to be necessarily gained by expensive experiments. In Chapter 6 "Data-driven computations with RVE generated data sets" we investigate the possibility of deriving data from numerical experiments. Therefore, we use a representative volume element of a foam material. The material behavior of the full matrix material is known and with the knowledge we can conduct simulations with the volume element. Six simple simulations with the representative volume element are needed to generate a full data set for describing the linear behavior of the foam. Afterwards, we conduct a full three-dimensional simulation with the generated data and the numerical improvement of the prior chapter.

Chapter 7 "Examples of applications" focuses on further applications of the data-driven finite element method. As a method that can insert an uncertain behavior of the material we use it here for a polymorphic approach. Furthermore, we apply the data-driven framework to a diffusion equation here. The data-driven method makes it possible to

replace the constitutive law, not only in mechanical equations but also in other field equations such as electric, heat or diffusion problems. Lastly, in Chapter 8 “Conclusions”, we conclude the thesis and take a look at the future of the data-driven finite element method.

2 Governing equations

As a new method, the data-driven approach is introduced to a relatively simple framework of the finite element analysis. By using the same energy densities as the linear finite element analysis, it is very similar to it, although it does not restrict to linear material data. To compare the data-driven finite element method (DD-FEM) with the classical finite element method (classical FEM) and the stochastic finite element method (SFEM), we briefly recap basic definitions and assumptions of linear elasticity and then move on to the different finite element methods. We will derive the basic equations for the classic, stochastic and data-driven approaches in this chapter and compare the equations, which need to be solved numerically. Further information and more detailed derivations may be found in one of the many literatures towards the finite element analysis, like the well-known books of Hughes [51] and Zienkiewicz [117, 118], or for example in [111, 119]. A well-examined introduction to the stochastic finite element analysis is given by Papadopoulos [92], or Der Kiureghian [32], while for an introduction into the new DD-FEM we refer the works of Kirchdoerfer [66, 67].

2.1 The boundary value problem of linear elasticity

To describe the motion of a body Ω in a continuum mechanic problem, the main variable of interest is the displacement field $\boldsymbol{u}(\boldsymbol{x}, t) : \Omega \times [0, T] \to \mathbb{R}^3$. It characterizes the motion of the body between the reference configuration and the actual configuration. Given a kinematic and a constitutive equation we are able to derive the strains and stresses in the body from the displacement. In linear elasticity, there are three main assumptions made, these are:

- small deformations,
- a linear constitutive equation,
- equilibrium constraint at the undeformed body.

Presuming small deformations, that is $||\boldsymbol{u}|| \ll 1$ and $||\boldsymbol{\nabla u}|| \ll 1$, we can write the strain field $\boldsymbol{\epsilon}(\boldsymbol{u})$ as a linearized function of the displacement field. Neglecting higher order terms, the strain tensor is the symmetrized gradient of the displacement field $\boldsymbol{u}$

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right) =: \boldsymbol{\nabla}_{\text{sym}}(\boldsymbol{u}) \tag{2.1}$$

where the strain tensor is a second order tensor with components

$$\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{pmatrix}.$$

The strain is connected by the constitutive equation to the stress tensor $\boldsymbol{\sigma}$. By a linear constitutive equation, known as Hooke's law, we have

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \mathbb{C} : \boldsymbol{\epsilon} \tag{2.2}$$

where $\mathbb{C}$ is the fourth-order material stiffness tensor, which is solely responsible for describing the material behavior. Because of the symmetries of the stress and strain tensor and assuming isotropic material behavior, the stiffness tensor can be described by two independent values

$$C_{ijkl} = \frac{E\nu}{(1+\nu)(1-2\nu)}\delta_{ij}\delta_{kl} + \frac{E}{2(1+\nu)}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\,. \tag{2.3}$$

Commonly, those are the Young's or elastic modulus E and the Poisson's ratio ν while δ_{ij} here denotes the Kronecker delta, which is only one if $i = j$ and zero otherwise. Two other parameters used to describe the stiffness tensor in this work are the Lamé parameters

$$\lambda = \frac{\nu}{1-2\nu} \cdot \frac{E}{1+\nu}\,,$$
$$\mu = \frac{1}{2} \cdot \frac{E}{1+\nu}$$

which means we can simply convert both representations into each other. In general, material properties vary from point to point in space and in time. They can be dependent on several external effects such as temperature or pressure, but also on inner aspects such as impurities or gas inclusions. Therefore, we should keep in mind that choosing material parameters for a complete structural component in a numerical simulation is always a simplification of reality. In this context, we will later compare the data-driven and the stochastic method with each other, which can account for material uncertainties. To account for this uncertainty the stochastic finite element method adds a stochastic term to the stiffness tensor, and the data-driven finite element method includes material fluctuations in the data.

To describe the entire mechanic problem we need, besides the kinematic and the constitutive, a balance equation that puts outer and inner forces in balance. That is the

balance of linear momentum: For every point $\boldsymbol{x}$ of a domain $\Omega \subset \mathbb{R}^3$ it holds

$$\operatorname{div}(\boldsymbol{\sigma}) + \boldsymbol{b} = \rho \ddot{\boldsymbol{u}}\,, \tag{2.4}$$

where $\boldsymbol{b}(\boldsymbol{x})$ denotes the body force per unit volume, $\rho(\boldsymbol{x})$ is the material density and $\ddot{\boldsymbol{u}}(\boldsymbol{x})$ is the acceleration which is set to zero here. The kinematic, constitutive and balance equations fundamentally define the boundary volume problem of linear elasticity. For the complete definition, however, we are missing the boundary conditions

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \Gamma_u \tag{2.5}$$

$$\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n} \text{ on } \Gamma_\sigma \tag{2.6}$$

where Γ_u and Γ_σ are the disjoint sets of the Dirichlet and Neumann boundaries, e.g. $\Gamma_u \cup \Gamma_\sigma = \partial\Omega$ and $\Gamma_u \cap \Gamma_\sigma = \emptyset$. Summarizing the prior equations, we have the differential form (strong form) of the boundary value problem in linear elasticity:

Find $\boldsymbol{u}$ such that the balance of linear momentum (2.4) is fulfilled given

(2.1) the kinematic equation $\boldsymbol{\epsilon}(\boldsymbol{u}) = \boldsymbol{\nabla}_{\text{sym}}(\boldsymbol{u})$,

(2.2) Hooke's law $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\epsilon}$,

(2.5) the Dirichlet boundary $\boldsymbol{u} = \bar{\boldsymbol{u}}$ and

(2.6) the Neumann boundary $\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$

2.2 Linear finite element analysis

To solve such boundary value problems, the finite element method established itself in the last decades. Used for a variety of applications this method is generally accepted and should be well-known by most mechanical engineers. More than 70 years ago Courant [28] recognized that a decomposition of the domain can be beneficial for gaining a solution. These 70 years later, there exist a lot of commercial software tools that include computer-aided design possibilities that can solve problems from most mechanical engineering fields. As there is a lot of literature, it is also referred to in [51, 117, 118, 41] for more information.

The weak form

To solve the given differential problem via the finite element method we derive the weak form of the boundary value problem. There are several possibilities to do this and we will not use the balance of linear momentum (2.4) directly. We choose an energetic approach as it serves for a better comparison to the data-driven approach, which is also derived by an energetic approach. In the appendix, it is shown that we end up with the same variational problem if we start at the balance equation. In order to find the deformation of the solid, the principle of minimum potential energy states that the body takes the configuration, which minimizes its total potential energy. The potential energy is defined by the interior and exterior energies

$$\Pi(\boldsymbol{u}) = \Pi^{\text{int}}(\boldsymbol{u}) - \Pi^{\text{ex}}(\boldsymbol{u}) \tag{2.7}$$

where the internal energy is the integral over the elastic energy density

$$\Psi^{\text{el}}(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\epsilon} : \mathbb{C} : \boldsymbol{\epsilon} = \frac{1}{2}\boldsymbol{\epsilon} : \boldsymbol{\sigma} \tag{2.8}$$

and therefore

$$\Pi^{\text{int}} = \int_{\Omega} \Psi^{\text{el}}(\boldsymbol{\epsilon}) \, \mathrm{d}\Omega \, .$$

The term Π^{ex} in (2.7) summarizes the externally applied forces with the traction and the body forces

$$\Pi^{\text{ex}} = \int_{\Omega} \boldsymbol{b} \cdot \boldsymbol{u} \, \mathrm{d}\Omega + \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \boldsymbol{u} \, \mathrm{d}\Gamma \, . \tag{2.9}$$

Plugging in both energies the total potential energy is

$$\Pi = \frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon} : \mathbb{C} : \boldsymbol{\epsilon} - \int_{\Omega} \boldsymbol{b} \cdot \boldsymbol{u} \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \boldsymbol{u} \, \mathrm{d}\Gamma$$

and the variation of Π with respect to $\boldsymbol{u}$ then reads

$$\delta\Pi = \int_{\Omega} \delta\boldsymbol{\epsilon} : \boldsymbol{\sigma} \, \mathrm{d}\Omega - \int_{\Omega} \boldsymbol{b} \cdot \delta\boldsymbol{u} \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \delta\boldsymbol{u} \, \mathrm{d}\Gamma \tag{2.10}$$

which needs to vanish for a minimum.

The weak form of the problem then reads as:

Solve

$$\delta\Pi = \int_\Omega \delta\boldsymbol{\epsilon} : \boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{b}\cdot\delta\boldsymbol{u}\,\mathrm{d}\Omega - \int_{\Gamma_\sigma} \boldsymbol{t}\cdot\delta\boldsymbol{u}\,\mathrm{d}\Gamma = 0$$

with respect to

(2.1) the kinematic equation $\boldsymbol{\epsilon}(\boldsymbol{u}) = \boldsymbol{\nabla}_{\mathrm{sym}}(\boldsymbol{u})$,

(2.5) the Dirichlet boundary $\boldsymbol{u} = \bar{\boldsymbol{u}}$.

The two constraints which we are missing from the strong form are now built in the main equation. That solving $\delta\Pi = 0$ actually leads to a minimum is shown in the appendix. Equation (2.10) is also called the variational formulation of the elastic problem, where we start with the typical finite element approximations next.

Finite element discretization

The goal is to find the minimum of the total elastic energy. To find the minimum, numerical help is needed. In the following, we introduce some finite element notations that allow us to take the step from the theoretical equations to the discretized problem as a system of linear equations that we can then solve.
The key aspect of the finite element method is the decomposition of the domain Ω in smaller subdomains Ω_e which are called finite elements. For the finite elements often regular and simple geometries are used, such that the union approximates the real geometry. Also, the finite elements yield a disjoint decomposition of the domain, that is $\Omega_i \cap \Omega_j = \emptyset \quad \forall\, i, j \in \{1, ..., n_e\}, i \neq j$ and

$$\Omega \approx \tilde{\Omega} = \bigcup_{\mathcal{E}} \Omega_e = \bigcup_{e=1}^{n_e} \Omega_e. \tag{2.11}$$

The set $\mathcal{E}$ is the set of all elements $\mathcal{E} = \{1, \ldots, n_e\}$. Using Voigt's notation simplifies many of the following equations. For the stiffness, stress and strain tensor we use from now on

$$\mathbb{C} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{pmatrix}, \qquad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{xz} \\ \epsilon_{xy} \end{pmatrix}, \qquad \boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix}.$$

Hooke's law (2.2) is then the matrix vector multiplication

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\epsilon}\,. \tag{2.12}$$

The displacement field $\boldsymbol{u}$ as well as its variation are approximated by a weighted sum of shape functions $N_1, ..., N_{n_k}$

$$\boldsymbol{u}(\boldsymbol{x}) \approx \tilde{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{k=1}^{n_k} N_k(\boldsymbol{x})\hat{\boldsymbol{u}}_k = \mathbf{N}\cdot\hat{\boldsymbol{u}} \tag{2.13}$$

$$\delta\tilde{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{k=1}^{n_k} N_k(\boldsymbol{x})\hat{\boldsymbol{v}}_k = \mathbf{N}\cdot\hat{\boldsymbol{v}} \tag{2.14}$$

where $\mathbf{N}$ is the vector of the shape functions and the coefficient vectors $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$ contain the unknown weights. More about the choice of shape functions and their properties is found in the FEM implementation Section 3.1. Defining the differential matrix $\mathbf{B}$ by its element-wise part

$$\mathbf{B}^e = \begin{pmatrix} N_{1,x} & 0 & 0 & & N_{n_k,x} & 0 & 0 \\ 0 & N_{1,y} & 0 & & 0 & N_{n_k,y} & 0 \\ 0 & 0 & N_{1,z} & \dots & 0 & 0 & N_{n_k,z} \\ 0 & N_{1,z} & N_{1,y} & & 0 & N_{n_k,z} & N_{n_k,y} \\ N_{1,z} & 0 & N_{1,x} & & N_{n_k,z} & 0 & N_{n_k,x} \\ N_{1,y} & N_{1,x} & 0 & & N_{n_k,y} & N_{n_k,x} & 0 \end{pmatrix} \tag{2.15}$$

where $N_{i,j}$ is the derivative of the i-th shape function with respect to j and n_k is the number of shape functions per element, we can get the element-wise strains by the matrix vector calculation

$$\boldsymbol{\epsilon}^e = \mathbf{B}^e\hat{\boldsymbol{u}}^e\,. \tag{2.16}$$

We obtain the stresses by multiplying the strains with the stiffness matrix in Voigt's notation

$$\boldsymbol{\sigma}^e = \mathbb{C}\boldsymbol{\epsilon}^e = \mathbb{C}\mathbf{B}^e\hat{\boldsymbol{u}}^e\,. \tag{2.17}$$

Now, if we use (6.19), (2.14), (2.16) and (2.17) and plug it into the variation of the total potential energy (2.10) we get

$$\begin{aligned} \delta\Pi &= \hat{\boldsymbol{v}}^T\cdot\int_\Omega \mathbf{B}^T\mathbb{C}\mathbf{B}\,\mathrm{d}\Omega\cdot\hat{\boldsymbol{u}} - \hat{\boldsymbol{v}}^T\cdot\int_\Omega \boldsymbol{b}\cdot\mathbf{N}\,\mathrm{d}\Omega - \hat{\boldsymbol{v}}^T\cdot\int_{\Gamma_\sigma}\boldsymbol{t}\cdot\mathbf{N}\,\mathrm{d}\Gamma = 0 \\ \Leftrightarrow\ \hat{\boldsymbol{v}}^T\cdot\underbrace{\int_\Omega \mathbf{B}^T\mathbb{C}\mathbf{B}\,\mathrm{d}\Omega}_{\mathbf{K}}\cdot\hat{\boldsymbol{u}} &= \hat{\boldsymbol{v}}^T\cdot\underbrace{\left(\int_\Omega \boldsymbol{b}\cdot\mathbf{N}\,\mathrm{d}\Omega + \int_{\Gamma_\sigma}\boldsymbol{t}\cdot\mathbf{N}\,\mathrm{d}\Gamma\right)}_{\mathbf{f}} \end{aligned}$$

such that we get the typical FEM system

$$\mathbf{K} \cdot \widehat{\boldsymbol{u}} = \mathbf{f} \tag{2.18}$$

where $\mathbf{K}$ is the stiffness matrix

$$\mathbf{K} = \int_\Omega \mathbf{B}^T \mathbb{C} \mathbf{B} \, \mathrm{d}\Omega \tag{2.19}$$

and the load vector $\mathbf{f}$ summarizes the body forces $\boldsymbol{b}$ and traction forces $\boldsymbol{t}$

$$\mathbf{f} = \int_\Omega \mathbf{N}^T \boldsymbol{b} \, \mathrm{d}\Omega + \int_{\Gamma_\sigma} \mathbf{N}^T \boldsymbol{t} \, \mathrm{d}\Gamma \,. \tag{2.20}$$

The element-wise notation reads

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT} \mathbb{C} \mathbf{B}^e \, \mathrm{d}\Omega \qquad\qquad \mathbf{K} = \bigcup_{\mathcal{E}} \mathbf{K}^e$$

$$\mathbf{f}^e = \int_{\Omega^e} \mathbf{N}^{eT} \boldsymbol{b}^e \, \mathrm{d}\Omega + \int_{\Gamma_\sigma^e} \mathbf{N}^{eT} \boldsymbol{t}^e \, \mathrm{d}\Gamma \qquad\qquad \mathbf{f} = \bigcup_{\mathcal{E}} \mathbf{f}^e$$

where $\bigcup_{\mathcal{E}}$ denotes the assembly of element-wise computations

$$\int_\Omega \bullet \, \mathrm{d}\Omega = \bigcup_{\mathcal{E}} \int_{\Omega^e} \bullet \, \mathrm{d}\Omega \,.$$

Remarks

1. In order to distinguish the different finite element methods, we will call simulations obtained by means of the previously described method with a determined material law *classical* FEM.

2. It is also possible to derive the solution by considering the dual energy density, see [119] for example. The complementary energy density as a function of the stress follows from a Legendre transform, $\Psi^*(\boldsymbol{\sigma}) = \max_{\boldsymbol{\epsilon}} \left[\boldsymbol{\sigma} : \boldsymbol{\epsilon} - \Psi^{\mathrm{el}}(\boldsymbol{\epsilon}) \right]$, to

$$\Psi^*(\boldsymbol{\sigma}) = \frac{1}{2} \boldsymbol{\sigma} : \mathbb{D} : \boldsymbol{\sigma} \tag{2.21}$$

   where $\mathbb{D}$ is the compliance tensor. It is then called the principle of minimum complementary potential energy and the problem to solve reads as:

Minimize the conjugated elastic potential

$$\Pi = \frac{1}{2} \int_\Omega \boldsymbol{\sigma} : \mathbb{D} : \boldsymbol{\sigma} - \int_{\Gamma_u} \boldsymbol{t} \cdot \bar{\boldsymbol{u}} \, \mathrm{d}\Omega$$

with the side constraints

(2.4) the balance of linear momentum $\operatorname{div}(\boldsymbol{\sigma}) - \boldsymbol{b} = 0$ and

(2.6) the Neumann boundary $\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$

The elastic and its conjugated principle are called dual as the constraints of the elastic energy are in the essential equation of the conjugated and the other way round. The elastic energy density Ψ^{el} as well as the conjugate energy density Ψ^* will be used in the data-driven method. The elastic one as a distance measure in the strain space and the conjugate as a distance measure in the stress space.

2.3 Stochastic finite element analysis

Approaches to take uncertainties of model input parameters into account go back to the early 1970s and grew significantly since the 1990s due to the permanent increase of computing power. Well-established stochastic methodologies were introduced to engineering systems to quantify the uncertainty [92]. Applied to mechanical problems, the introduced methods are assigned to computational stochastic mechanics, which is a small but prolific scientific field [92, 33, 85]. In the stochastic finite element method (SFEM) stochastic variables replace their deterministic counterparts and it can therefore be seen as an extension of the deterministic classical FEM. By replacing the deterministic variables the SFEM still uses all advantages of the finite element analysis as a versatile and valuable tool but can now be used for the simulation with uncertainties. The origin of the uncertainty is not important. It is possible to add a stochastic variable in the material (Young's Modulus, Poisson's Ratio, yield stress, ...), the geometry (fluctuation of nominal size) or the loading (fluctuations of the applied force). The field of stochastic processes also yields the possibility to describe the behavior of those variables in space and time, however, often only one or two uncertainties are considered and a time-dependence is omitted too.

From the users point of view two things change in contrast to the classical finite element analysis. First, the stochastic input has to be characterized. Knowledge about the variability of a certain variable is usually gained by multiple measurements. In many situations, however, it is difficult to conduct so many measurements that distributions can be derived. Second, using stochastic variables renders the differential equation now also stochastic such that the solution is also a stochastic variable now. Therefore, we

are interested in the distribution or the statistics, e.g. mean and variance, of the displacement instead of a deterministic value. The applications of the SFEM are manifold [10, 47]; it has been used in a large number of problems like solid and fluid mechanics, acoustics, heat transfer and biomechanics, [17, 37, 42, 99]. The paper of Arregui-Mena [10] shows an overview of 17 practical case studies of uncertainties in polymer nanocomposites, porous materials, metal foams, total knee replacements and geomaterials where the SFEM is used.

The stochastic differential equation

As we want to compare the SFEM to the DD-FEM the only uncertainty we consider here will be material uncertainties. Voids, impurities, volume shrinkage or other imperfect conditions may influence the material parameters. In the most general notation we have a stochastic process that depends on space, time and randomness

$$\chi : \Omega \times [0, T] \times \mathbb{P} \to \mathbb{R}^n$$

where $\mathbb{P}$ is a probability space. For every element p of the probability space we have a function of space and time

$$p \mapsto \chi_p(\boldsymbol{x}, t)\,.$$

As mentioned above we will have no change in time. One way to consider material deviations is to superpose a zero-mean stochastic variable $\chi_p(\boldsymbol{x})$ to the strain-stress relation, i.e.

$$\mathbb{C}_\chi(\boldsymbol{x}) = \mathbb{C}^\mu(\boldsymbol{x})\Big(1 + \chi_p(\boldsymbol{x})\Big). \tag{2.22}$$

Here $\mathbb{C}^\mu$ is the local mean elasticity tensor and the stochastic term

$$1 + \chi_p(\boldsymbol{x}) \tag{2.23}$$

describes a percentage change around the 'true' value. The elastic energy (2.8) now changes to

$$\Psi^{\mathrm{el}}_\chi(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\epsilon} : \mathbb{C}^\mu\Big(1 + \chi\Big) : \boldsymbol{\epsilon}\,. \tag{2.24}$$

For the evaluation of the stochastic finite element equations the modified elastic energy (2.24) needs to be used in the same manner now as in the classic FEM evaluation. Therefore, only the stiffness matrix changes. Simple calculations result in the finite

element stiffness matrix for the SFEM

$$\mathbf{K} = \int_\Omega \mathbf{B}^T \mathbb{C}^\mu \Big(1 + \chi\Big) \mathbf{B} \, \mathrm{d}\Omega \,. \tag{2.25}$$

Hence the stiffness matrix can be split into a deterministic part using the mean elastic parameter and a stochastic part

$$\mathbf{K} = \underbrace{\int_\Omega \mathbf{B}^T \mathbb{C}^\mu \mathbf{B} \, \mathrm{d}\Omega}_{\mathbf{K}^\mu} + \underbrace{\int_\Omega \mathbf{B}^T \chi \mathbb{C}^\mu \mathbf{B} \, \mathrm{d}\Omega}_{\mathbf{K}^\chi} \tag{2.26}$$

and the following SFEM system of equations needs to be solved, cf. [92]

$$\left(\mathbf{K}^\mu + \mathbf{K}^\chi\right) \widehat{\boldsymbol{u}} = \mathbf{f} \tag{2.27}$$

As small as the change in the equations may be, the stochastic finite element method involves some difficulties that are hard to overcome. The stochastic structure changes the way of solving (2.27) significantly. To determine the stochastic structure of the response variable a single computation of the system is not sufficient anymore. To solve the stochastic differential equation several methods have been developed. A detailed classification can be found in [105]. In the simplest case this is done by a Monte Carlo simulation, which is a large number of simulations with realizations of the stochastic variables. Using a realization of the stochastic part the problem (2.27) renders again deterministic and can be solved directly as before. However, to account for the stochastic distribution, a large number of computations needs to be carried out then. This is numerically very costly but also yields an approximation of the response distribution instead of only statistics. Other methods like perturbation-based methods or projection and decomposition methods have been developed to accelerate the computation, see [92, 104, 105]. For the applied stochastic field and the solving method used in this thesis see Section 3.2.

2.4 Data-driven finite element analysis

Last, we present the data-driven finite element method. It was first proposed in [67] by Trenton Kirchdoerfer and Michael Ortiz in 2016 and is, therefore, a relatively new method. They derived the method for truss structures and linear elasticity. Even though we will show some examples in the truss case we focus on the derivation for the three-dimensional case. The derivation and steps are the same for trusses only with the lower dimensional variables and can also be found in [67] and [66].

The main idea is to use a data set $\mathcal{D}$ that is gained in experiments directly without an empirical treatment. Typically the empirical treatment follows up the experiment by

fitting appropriate material models to the data. Now, the information about the material is used as it is gained in the experiment. Measured values are neither averaged nor is information gained in areas of no measurements. The values go into the algorithm in their original state. The fitting of the material model is skipped, which means that modeling errors are prevented. However, the characteristic of the boundary value problem changes significantly as there is no interception of the physical equilibrium and the material equation anymore. To make this clear, consider the left image of Fig. 2.1. We find the equilibrium of an arbitrary body as the intersection of the material law and the physical constraints. However, on the right side, where we have the strain-stress relationship only as a set of data points, an intersection does not need to exist nor is very likely to exist at any time. Therefore, instead of the intersection, we search for the closest distance between the data set $\mathcal{D}$ and the constraint set $\mathcal{C}$. A minimum distance problem arises then and needs to be solved.

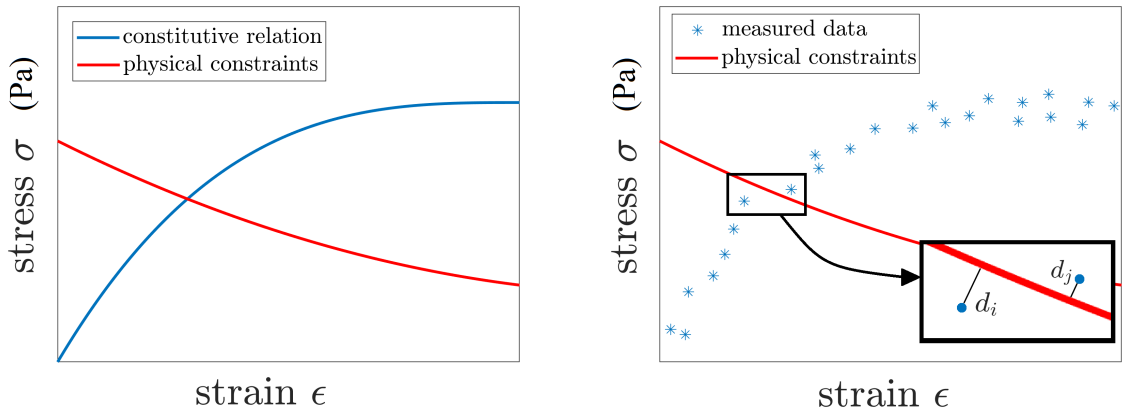


Figure 2.1: Difference of having a functional constitutive model (left) in contrast to a data set as constitutive model (right).

In the following, we introduce the DD-FEM theory, which mainly coincides with the description in [67, 66] by Kirchdoerfer and Ortiz.

Minimizing distance data-driven finite element method

The task is to find an appropriate norm in which we can measure the distance between the data set and the equilibrium constraint set. Different units do not allow us to take the distance directly in Fig. 2.1. The energy densities (2.8) and (2.21) are suitable but, as the idea of the data-driven method is to elide any material law in functional form, the material tensors $\mathbb{C}$ and $\mathbb{D}$ are unknown. Therefore, we cannot make use of the classical

energy densities but instead replace them by pure numerical tensors $\mathbb{C}^\circ$ and $\mathbb{D}^\circ$

$$\Psi_\circ^{\mathrm{el}}(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\epsilon} : \mathbb{C}^\circ : \boldsymbol{\epsilon} \tag{2.28}$$

and

$$\Psi_\circ^{*}(\boldsymbol{\sigma}) = \frac{1}{2}\boldsymbol{\sigma} : \mathbb{D}^\circ : \boldsymbol{\sigma} \tag{2.29}$$

to define appropriate energy densities. The constitutive matrices $\mathbb{C}^\circ$ and $\mathbb{D}^\circ$ are here of purely numerical nature, i.e., they have to guarantee the basic property of symmetry but are arbitrary values otherwise. The idea is now to find the closest-to-the-real $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$-pair to identify the deformation. This data pair is expected to minimize the sum of the prior two energy densities of the difference to the real state. Accordingly, we formulate a local penalty function

$$\Psi_\circ(\boldsymbol{\epsilon}, \boldsymbol{\sigma}) = \min_{(\boldsymbol{\epsilon}^i, \boldsymbol{\sigma}^i) \in \mathcal{D}} \left(\Psi_\circ^{\mathrm{el}}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^i) + \Psi_\circ^{*}(\boldsymbol{\sigma} - \boldsymbol{\sigma}^i) \right) \tag{2.30}$$

which defines the minimal distance of the data set $\mathcal{D}$ to a given strain-stress pair by the norm induced through (2.28) and (2.29). This also shows the purpose of the matrices $\mathbb{C}^\circ$ and $\mathbb{D}^\circ$. They serve as a weight to guarantee that the strain and stress parts in (2.30) are in the same order of magnitude and get the same unit. To enforce this minimization at every point positivity of the energy densities allows us to use the integral over the whole domain of the body such that the global penalty function can be defined by

$$W^{\mathrm{py}} = \int_\Omega \Psi_\circ(\boldsymbol{\epsilon}, \boldsymbol{\sigma}) \, \mathrm{d}\Omega. \tag{2.31}$$

This function needs to be minimized with respect to the physical and boundary constraints and the data-driven problem can then be summarized as follows:

Minimize the global penalty function

$W^{\mathrm{py}} = \int_\Omega \Psi_\circ(\boldsymbol{\epsilon}, \boldsymbol{\sigma}) \, \mathrm{d}\Omega$

with respect to

(2.1) the kinematic equation $\boldsymbol{\epsilon}(\boldsymbol{u}) = \boldsymbol{\nabla}_{\mathrm{sym}}(\boldsymbol{u})$,

(2.4) the balance of linear momentum $\mathrm{div}(\boldsymbol{\sigma}) + \boldsymbol{b} = 0$,

(2.5) the Dirichlet boundary $\boldsymbol{u} = \bar{\boldsymbol{u}}$ and

(2.6) the Neumann boundary $\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$.

Boundary conditions aside, we first deal with the kinematic and equilibrium ones. The

kinematic side constraint can be enforced easily by plugging the kinematic equation (2.1) into (2.31). The equilibrium constraint, however, needs to be enforced otherwise. We introduce here a Lagrange multiplier $\boldsymbol{\lambda}$ to enable (2.4). It is also possible to choose a different technique to enforce the second side constraint, for example by introducing a penalty-like variable. Adding the Lagrange multiplier we end up with the mixed energy functional

$$W^* = \int_\Omega \Psi_\circ(\boldsymbol{\epsilon}, \boldsymbol{\sigma}) \, \mathrm{d}\Omega - \int_\Omega \boldsymbol{\lambda} \cdot (\mathrm{div}(\boldsymbol{\sigma}) + \boldsymbol{b}) \, \mathrm{d}\Omega \,. \tag{2.32}$$

Using the standard finite element steps of integration by parts, the approximation of the displacement and the same approach for the Lagrangian and plugging in the finite element representation for the strains, we get the stationary problem

$$\delta \left[\int_\Omega \Psi_\circ \left((\mathbf{B}\widehat{\boldsymbol{u}}, \boldsymbol{\sigma}) \right) \mathrm{d}\Omega + \widehat{\boldsymbol{\lambda}}^T \left(\int_\Omega \mathbf{B}^T \boldsymbol{\sigma} \, \mathrm{d}\Omega - \mathbf{f} \right) \right] = 0$$

where $\boldsymbol{f}$ inhabits the force terms of the body and the Neumann boundary. Using (2.30) with $\mathbb{D}^\circ = (\mathbb{C}^\circ)^{-1}$ and $\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*$ as the optimal data points we can write

$$\delta \left[\frac{1}{2} \int_\Omega \left((\mathbf{B}\widehat{\boldsymbol{u}} - \boldsymbol{\epsilon}^*)^T \, \mathbb{C}^\circ (\mathbf{B}\widehat{\boldsymbol{u}} - \boldsymbol{\epsilon}^*) + (\boldsymbol{\sigma} - \boldsymbol{\sigma}^*)^T (\mathbb{C}^\circ)^{-1} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) \right) \mathrm{d}\Omega + \right.$$
$$\left. \widehat{\boldsymbol{\lambda}}^T \left(\int_\Omega \mathbf{B}^T \boldsymbol{\sigma} \, \mathrm{d}\Omega - \mathbf{f} \right) \right] = 0$$

Taking the variations with respect to the displacement, stress and Lagrangian we get:

$$\delta \widehat{\boldsymbol{u}} \Rightarrow \int_\Omega \mathbf{B}^T \mathbb{C}^\circ \left(\mathbf{B}\hat{\mathbf{u}} - \boldsymbol{\epsilon}^* \right) \mathrm{d}\Omega = 0 \tag{2.33}$$

$$\delta \boldsymbol{\sigma} \Rightarrow \int_\Omega (\mathbb{C}^\circ)^{-1} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) \, \mathrm{d}\Omega + \int_\Omega \mathbf{B}\widehat{\boldsymbol{\lambda}} \, \mathrm{d}\Omega = 0 \tag{2.34}$$

$$\delta \widehat{\boldsymbol{\lambda}} \Rightarrow \int_\Omega \mathbf{B}^T \boldsymbol{\sigma} \, \mathrm{d}\Omega - \mathbf{f} = 0. \tag{2.35}$$

Equation (2.33) we rewrite to

$$\int_\Omega \mathbf{B}^T \mathbb{C}^\circ \mathbf{B} \, \mathrm{d}\Omega \; \widehat{\boldsymbol{u}} = \int_\Omega \mathbf{B} \mathbb{C}^\circ \boldsymbol{\epsilon}^* \, \mathrm{d}\Omega \tag{2.36}$$

which leads to a system of equations for the unknown displacement values $\widehat{\boldsymbol{u}}$. The second

equation (2.34) implies

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^* - \mathbb{C}^\circ \mathbf{B} \widehat{\boldsymbol{\lambda}} \tag{2.37}$$

which can be plugged into the third equation then. The unknown stresses are replaced by the unknown Lagrangian $\widehat{\boldsymbol{\lambda}}$

$$\int_\Omega \mathbf{B}^T \mathbb{C}^\circ \mathbf{B} \, \mathrm{d}\Omega \; \widehat{\boldsymbol{\lambda}} = \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}^* \, \mathrm{d}\Omega - \mathbf{f} \tag{2.38}$$

to get a second FEM equation for the Lagrangian. Summarizing the prior equations, we can write the two DD-FEM linear equation systems as

$$\mathbf{K}_u \widehat{\boldsymbol{u}} = \mathbf{f}_u \tag{2.39}$$

$$\mathbf{K}_\lambda \widehat{\boldsymbol{\lambda}} = \mathbf{f}_\lambda \tag{2.40}$$

with

$$\mathbf{K}_u = \int_\Omega \mathbf{B}^T \mathbb{C}^\circ \mathbf{B} \, \mathrm{d}\Omega \qquad \mathbf{f}_u = \int_\Omega \mathbf{B} \mathbb{C}^\circ \boldsymbol{\epsilon}^* \, \mathrm{d}\Omega$$
$$\mathbf{K}_\lambda = \int_\Omega \mathbf{B}^T \mathbb{C}^\circ \mathbf{B} \, \mathrm{d}\Omega \qquad \mathbf{f}_\lambda = \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}^* \, \mathrm{d}\Omega - \mathbf{f}$$

where we observe that $\mathbf{K}_{\boldsymbol{u}} = \mathbf{K}_\lambda$ and we can therefore name it $\mathbf{K}$. Inside the force vectors the values $\{\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*\}$ denote the still unknown optimal data points out of set $\mathcal{D}$. In Section 3.3 an iterative algorithm is proposed to compute them. The global systems are obviously gained by the assembly of the element-wise computations

$$\mathbf{K} = \bigcup_{\mathcal{E}} \mathbf{K}^e \qquad \mathbf{f}_u = \bigcup_{\mathcal{E}} \mathbf{f}_u^e$$
$$\mathbf{K} = \bigcup_{\mathcal{E}} \mathbf{K}^e \qquad \mathbf{f}_\lambda = \bigcup_{\mathcal{E}} \mathbf{f}_\lambda^e$$

Remarks

1. The use of the elastic energy densities (2.8) and (2.21) in (2.31) seems to restrict the problem to linear elasticity. This is not the case. The functions (2.8) and (2.21) provide a norm accounting for proper physical units and comparable stress and strain measures. In their first publication [67] Kirchdoerfer and Ortiz already used non-linear data. Also, by definition of (2.31) with the energy densities the basic principles of material theory hold, i. e. determinism, locality, and objectivity. An alternative norm would be the first or second invariant of the tensors for example. This choice, however, is more restrictive because it presumes isotropic tensor functions and thus presumes isotropic material. In case of an elasto-plastic

material response the choice (2.30) is also applicable.

2. To include different local behavior the data set, as well as the numerical tensor, can be made dependent on the location leading to $\mathcal{D}(x)$ and $\mathbb{C}^\circ(x)$. Transferred to the data-driven approach this means that the data set $\mathcal{D}$ and the numerical material tensor $\mathbb{C}^\circ$ can be made dependent on their element assignment $\mathcal{D}_e$ and $\mathbb{C}^\circ_e$, $e = 1, ..., n_e$. This does not affect the prior derivation. However, it allows to implement uncertainties or different material behaviors at different locations. Multiple materials or different data sets from the same material can be used to describe different local behavior. Going even further, in dynamic computations the data set can also be dependent on a time variable, leading to $\mathcal{D}^t_e$.
3. We want to mention that the definition of $\mathbb{C}^\circ$ and $\mathbb{D}^\circ$ corresponds to some basic assumptions on the model. Throughout this thesis we presume isotropic numerical stiffness and compliance tensors. The isotropy is here used for simplicity and better convergence of the data search but it is no requirement of the data-driven algorithm. Specifically we set $\mathbb{D}^\circ = (\mathbb{C}^\circ)^{-1}$ for all simulations.
4. Since the finite element procedure evaluates the given equations at the material points the data-driven algorithm needs to assign a data point to every material point, see also Section 3.3.
5. The dimension of the data points $(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)$ depends on the dimension of the problem. If we consider three-dimensional problems where the strain and stress tensor are described by six different values, the data points are twelve-dimensional. Two-dimensional problems result in six-dimensional data points and one-dimensional problems in two-dimensional data points $(\epsilon, \sigma) \in \mathbb{R}^2$. Therefore, typical DD-FEM properties will be illustrated by using a one-dimensional problem in the beginning of Section 4.1. For those, the data set is easily plotted, which allows an easier access to the algorithm and its properties.

Systems of equations

Concluding this chapter, we want to rewind the systems of equations, that we have to solve, to have a glance at them simultaneously. All three methods have the same structure of a system of equations $\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$ with a stiffness matrix $\mathbf{A}$, a force vector $\boldsymbol{y}$ and a solving variable $\boldsymbol{x}$ but with different inputs of $\mathbf{A}$, $\boldsymbol{x}$ and $\boldsymbol{y}$. The first difference, which can be seen in Table 2.1, is that in the data-driven setting two systems of equations have to be solved. However, the matrices coincide for both systems such that only one inversion of the matrix has to be computed.

The structure of the stiffness matrix (left-hand side) does not change significantly. The only difference between the methods is the material tensor. In the SFEM a stochastic variable is added while in the DD-FEM a numerical tensor as a weight is used. The

Method	Left-hand side	Solving variable	Right-hand side
classical FEM	$\int_\Omega \mathbf{B}^T \mathbb{C} \mathbf{B} \, \mathrm{d}\Omega$	$\widehat{\boldsymbol{u}}$	$\int_\Omega \mathbf{N}^T \boldsymbol{b} \, \mathrm{d}\Omega + \int_{\Gamma_\sigma} \mathbf{N}^T \boldsymbol{t} \, \mathrm{d}\Gamma$
SFEM	$\int_\Omega \mathbf{B}^T \mathbb{C}^\mu (1+\chi) \mathbf{B} \, \mathrm{d}\Omega$	$\widehat{\boldsymbol{u}}$	$\int_\Omega \mathbf{N}^T \boldsymbol{b} \, \mathrm{d}\Omega + \int_{\Gamma_\sigma} \mathbf{N}^T \boldsymbol{t} \, \mathrm{d}\Gamma$
DD-FEM	$\int_\Omega \mathbf{B}^T \mathbb{C}^\circ \mathbf{B} \, \mathrm{d}\Omega$	$\widehat{\boldsymbol{u}}$	$\int_\Omega \mathbf{B}^T \mathbb{C}^\circ \boldsymbol{\epsilon}^* \, \mathrm{d}\Omega$
		$\widehat{\boldsymbol{\lambda}}$	$\int\limits_\Omega \mathbf{B}^T \boldsymbol{\sigma}^* \, \mathrm{d}\Omega - \mathbf{f}$

Table 2.1: Comparison of the equations for the different finite element methods. The right-hand side of the first two methods equals $\mathbf{f}$ of the DD-FEM.

right-hand side of the classic FEM and the SFEM coincide as we do not use stochastic quantities in the applied forces. However, there is a bigger difference to the DD-FEM. The first system of equations for the displacements is driven by the strain data points, while the second system, for the Lagrangian, is driven by the out-of-balance forces with the stress data points where $\mathbf{f}$ inhabits the traction and body forces.

3 Numerical realization

After we introduced the theory of the three different finite element methods, we want to specify several numerical and technical details of the implementation and the execution of the numerical simulations. The stochastic, as well as the data-driven algorithm are implemented into an existing in-house finite element code in Matlab [5].
First, we state further theoretical details, such as the element type and the shape functions which are used for all three finite element methods. Then, for the stochastic approach, we state the random field that describes the material fluctuations as well as the solution method. For the data-driven method we specify the iterative algorithm for the optimal data points as well as the data set generation of the synthetic data which are used. At last, examples which will be considered in this thesis are introduced.

3.1 Finite element implementation

Ultimately, we have to solve the system of linear equations (2.18) for the classical FEM, (2.27) for the SFEM and (2.39) and (2.40) for the DD-FEM. Due to the decomposition of the domain

$$\Omega \approx \tilde{\Omega} = \bigcup_{\mathcal{E}} \Omega_e = \bigcup_{e=1}^{n_e} \Omega_e \tag{3.1}$$

the element-wise values need to be computed and assembled to establish the global system of equations. There are different possibilities to choose the elements which approximate the geometry. In this thesis, we apply linear hexahedral brick-type elements with 8 nodes throughout all simulations. We use an isoparametric approach where the geometry is approximated with the same shape functions as the displacements. On both sides of the equations derived in the prior chapter, integrals have to be solved in each element numerically. Using Gauss quadrature for the numerical integration we have

$$\int_{\Omega^e} f(x)\,\mathrm{d}x \approx \sum_{p=1}^{n_p} w_p f(x_p)$$

where w_p are the Gauss-weights and x_p are the integration points corresponding to Ω^e, which are also called material points. For the integration we use 8 material points

throughout all simulations. This value is particular, important for the data-driven approach. To every material point a data point needs to be assigned by a minimal distance evaluation. Depending on the number of data points in the data set, the search for the minimum distance can get a costly step of the data-driven computation. Therefore, increasing the material points especially increases the cost of the DD-FEM.

To do systematic calculations the physical elements of the body are transformed to a so called reference or master element. Common choices for the reference element are $[-1,1]^d$ or $[0,1]^d$ where d is the dimension. In the three-dimensional space the physical coordinates $\boldsymbol{x} = (x, y, z)$ describe the element before and the reference coordinates $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$ after the transformation. The coordinate transformation is then defined by

$$J_e = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \tag{3.2}$$

which is also depicted in Fig. 3.1. The integration points and shape functions are always

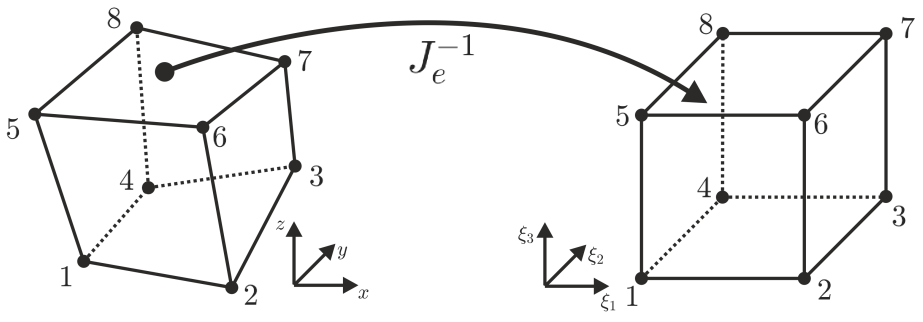


Figure 3.1: Transformation of the physical element to the reference element.

the same and only need to be computed once for the reference element. In the elements the displacement field is approximated by the shape functions N_i and the weights $\hat{\boldsymbol{u}}_i$

$$\boldsymbol{u} \approx \sum_{i=1}^{n_k} N_i \hat{\boldsymbol{u}}_i \,.$$

Common choices for the shape functions have the following two properties

$$N_i(\boldsymbol{\xi}^{(j)}) = \delta_{ij} \tag{3.3}$$

$$\sum_{i=1}^{n_k} N_i(\boldsymbol{\xi}) = 1 \tag{3.4}$$

where $\boldsymbol{\xi}^{(j)}$ are the coordinates of the j-th knot in the reference element. The collocation property states that the weights $\hat{\boldsymbol{u}}_i$ coincide with the displacement at the nodes. In this thesis linear Lagrange polynomials are used as shape functions in all simulations. They

are gained by a multiplicative approach for higher dimensions from their one-dimensional equivalents. Generally, the one-dimensional Lagrange polynomials of degree p read as follows

$$N_{i,p}(\xi) = \prod_{j=1, i\neq j}^{p+1} \frac{\xi^{(j)} - \xi}{\xi^{(j)} - \xi^{(i)}}$$

The linear Lagrange polynomials for the 1-D two node element are

$$N_1(\xi) = \frac{1}{2}(1+\xi) \qquad N_2(\xi) = \frac{1}{2}(1-\xi)$$

and the ones for three dimensions are then gained by the multiplicative approach

$$N_i(\xi_1, \xi_2, \xi_3) = N_j(\xi_1) N_k(\xi_2) N_l(\xi_3)$$

where $i = 1, ..., 8$ and $j, k, l \in \{1, 2\}$. This results in the eight functions

$$\begin{aligned}
N_1 &= \frac{1}{8}(1-\xi_1)(1-\xi_2)(1-\xi_3), & N_5 &= \frac{1}{8}(1-\xi_1)(1-\xi_2)(1+\xi_3) \\
N_2 &= \frac{1}{8}(1+\xi_1)(1-\xi_2)(1-\xi_3), & N_6 &= \frac{1}{8}(1+\xi_1)(1-\xi_2)(1+\xi_3) \\
N_3 &= \frac{1}{8}(1+\xi_1)(1+\xi_2)(1-\xi_3), & N_7 &= \frac{1}{8}(1+\xi_1)(1+\xi_2)(1+\xi_3) \\
N_4 &= \frac{1}{8}(1-\xi_1)(1+\xi_2)(1-\xi_3), & N_8 &= \frac{1}{8}(1-\xi_1)(1+\xi_2)(1+\xi_3)
\end{aligned}$$

of which everyone is assigned to a node of the element by relation (3.3). For this choice of shape functions the properties (3.3) and (3.4) both hold. The derivatives, which need to be derived for the $\mathbf{B}$ matrix, are omitted here due to the lengthy representation and easiness. For the numerical implementation they need to be evaluated at the integration points and used as in (2.15). Using Gauss quadrature, integral transformation and the shape functions we exemplarily write down the computation of the elemental stiffness matrix

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT} \mathbb{C} \mathbf{B}^e \, \mathrm{d}\Omega = \sum_{p=1}^{n_p} w_p \mathbf{B}^{eT}(\xi_p) \mathbb{C} \mathbf{B}^e(\xi_p) \det(J_e(\xi_p)).$$

which looks similar in all three cases. The only difference is the material matrix, which needs to be replaced by the methods equivalent. As an example we also have a look at the right-hand side of the first DD-FEM equation

$$\boldsymbol{f}_u^e = \int_{\Omega_e} \mathbf{B}^T \mathbb{C}^\circ \boldsymbol{\epsilon}^* \, \mathrm{d}\Omega = \sum_{p=1}^{n_p} w_p \mathbf{B}^{eT}(\xi_p) \mathbb{C}^\circ \boldsymbol{\epsilon}^*(\xi_p) \det(J_e(\xi_p))$$

and note that at every material point ξ_p a data point $(\boldsymbol{\epsilon}^*(\xi_p), \boldsymbol{\sigma}^*(\xi_p)) =: (\boldsymbol{\epsilon}^*_p, \boldsymbol{\sigma}^*_p)$ is needed. Therefore, to every material point a data point needs to be assigned, which describes the material state there. This is particularly important: To assign the optimal data point to the material point the data point, which is closest to the computed strains and stresses needs to be chosen. To find the closest data point a minimal distance search needs to be performed. This is, depending on dimension and number of data points, a numerically very expensive task. We focus more on that in Sections 3.3, 4.2 and Chapter 5. The corresponding stiffness matrices and force vectors of the other methods are gained analogously.

3.2 Stochastic finite element method

For the stochastic finite element method the random field and the solution method of the stochastic problem need to be defined. As we are looking for an overall comparison of the methods we use relatively simple methods here. A Gaussian noise is used as a random field and the Monte Carlo method as a solver. The Monte Carlo method can also be used in a certain sense for the random initializations in the data-driven computations later on.

Random fields

The random field is the important variable that transports the uncertainties in the SFEM. As always, the variable should be chosen as close to reality as possible. However, this is much more difficult for a stochastic process than for a single material constant. Like in one dimension random fields need to be characterized by parameters. As a consequence of the multi-dimensionality, the mean and variance are functions now which need to be determined by measurements in experiments. This is, however, a tough task. By construction of the stochastic term in (2.22) the mean function can be assumed to be zero. The covariance function specifies the correlation between two points, where the correlation should decrease as the distance increases in our case. Therefore, covariance functions are often dominated by a length scale parameter, which defines the degree of correlation with respect to the distance. To define this parameter we wish to know the material behavior at as many points of a material specimen as possible. Furthermore, we want to know them for as many different specimens as possible to deduce distributions. The problem of gaining this information, especially those inside a specimen, are hard or impossible to get. In [104] Stefanou states that, in general, assumptions have to be made to overcome the lack of experimental data. In [21] a sensibility study is run to characterize the influence of the parameters and to aid researchers in calibrating the random fields. In [11] digital image-based characterization, homogenization and representative volume element techniques are used as example.

One of the most common fields are Gaussian fields. Gaussian fields are technically very easy to handle and follow a normal distribution locally. The normal distribution assumption is quite frequent and also used if there is a lack of information. It occupies a special position in the stochastic distributions as the natural limit of sums of independent random variables. Stationary[1] Gaussian fields are fully characterized by its mean and covariance function. As described above the mean is set to zero. The options for the covariance functions are broad: rational, exponential or periodic are common, just to name a few. A more simple Gaussian process is the Gaussian noise with covariance function

$$C(\boldsymbol{x}, \boldsymbol{x}') = s^2 \delta_{\boldsymbol{x},\boldsymbol{x}'}$$

which is nonzero only for $\boldsymbol{x} = \boldsymbol{x}'$. Note that we are using s^2 for the variance and s for the standard deviation of the stochastic variables, as σ is already used throughout the thesis for the stresses. In Fig. 3.2 the left image is a Gaussian noise, while on the right side an exponential covariance function

$$C(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|}{l}\right)$$

with the length parameter l is demonstrated.

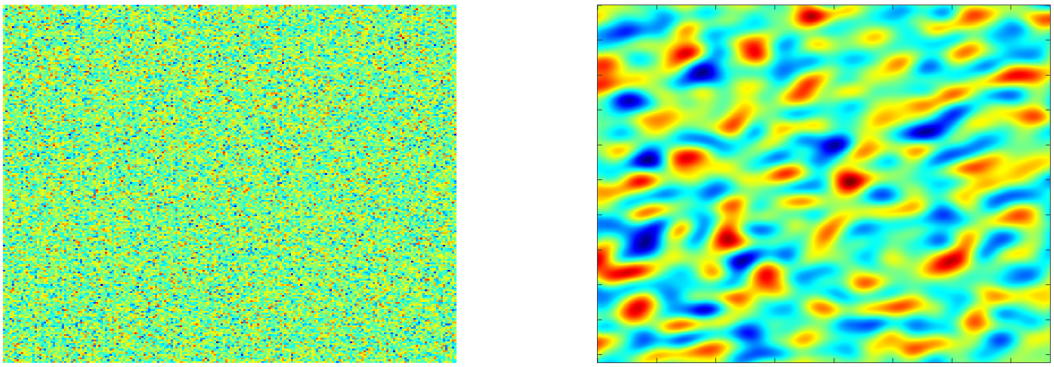


Figure 3.2: Two gaussian random fields which can indicate areas of higher (red) and lower (blue) elasticity. The first one is a Gaussian noise, the second uses an exponential covariance function.

The possibilities of stochastic processes are even wider. It is possible to describe more complex materials, for example such with a spatial anisotropy. Two examples for those

[1] Definition in the appendix.

fields are depicted in Fig. 3.3. The first image is an auto-correlated field with weighting in the perimeter, which is used to describe annual rings of wood [91]. In the second image we see an operator-scaling stable field as introduced in [9]. There, it is possible to define the preferential directions via the eigenvalues and eigenvectors of a scaling matrix. In the example which is plotted the eigenvalues $H_1^{-1} = 0.5, H_2^{-1} = 0.5$ and eigenvectors defined by the angles $\theta_1 = \exp(-0.3\pi i), \theta_2 = \exp(-0.2\pi i)$ are used. However, anisotropy goes

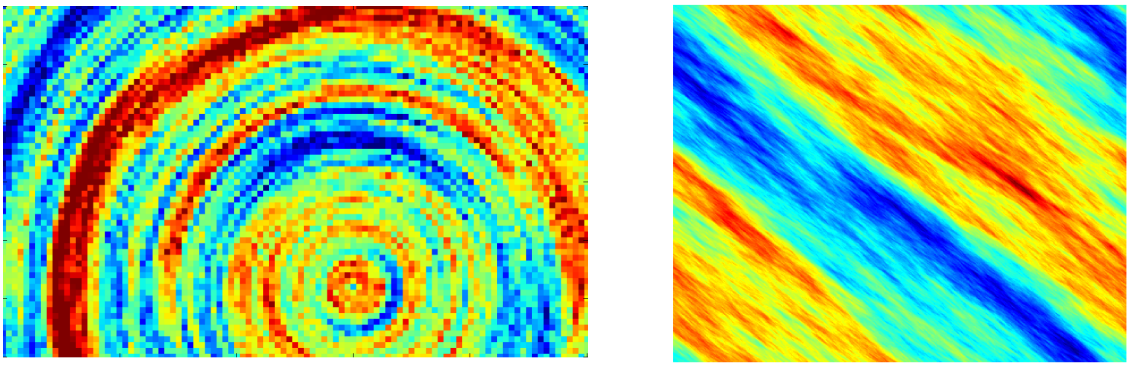


Figure 3.3: Two anisotropic stochastic random fields, which can indicate areas of higher (red) and lower (blue) elasticity. The first one is an auto-correlated field with weighting in the perimeter. The second image is an operator-scaling stable field where the eigenvectors define the orientation, see [9, 91]

beyond the scope of this thesis and we restrict ourselves to more easy processes such as the Gaussian noise. The Gaussian noise can be nicely compared to a data-driven solution with different data sets per element, which will be presented later. A broad introduction into common stochastic processes used in the SFEM can be found in [92].

The Gaussian noise

To simulate the material fluctuation we simulate a one-dimensional random field which specifies the percentage change in the Young's modulus. We can use construction (2.22) as we are able to extract the Young's modulus out of the constitutive matrix

$$(1 + \chi(\boldsymbol{x})) \cdot \mathbb{C}^{\mu} = (1 + \chi(\boldsymbol{x}))E \cdot \widehat{\mathbb{C}}^{\mu} \tag{3.5}$$

where $\widehat{\mathbb{C}}^{\mu} = 1/E \cdot \mathbb{C}^{\mu}$. Choosing a Gaussian noise means the values in space are uncorrelated and locally at every point we have a normal distribution. We use three different variances $s^2 \in \{0.1^2, 0.05^2, 0.01^2\}$ while the mean function is zero everywhere. Simple

linear transformations give us the distributions of the terms

$$\begin{aligned}\chi(\boldsymbol{x}) &\sim \mathcal{N}_{0,s^2}\\ 1+\chi(\boldsymbol{x}) &\sim \mathcal{N}_{1,s^2}\\ E(1+\chi(\boldsymbol{x})) &\sim \mathcal{N}_{E,E^2s^2}\end{aligned}$$

for a fixed but arbitrary $\boldsymbol{x}$. According to the 3-sigma rule of the normal distribution 99.7% of the fluctuations of the Young's modulus are then between $E+3Es$. With the prior values of s this leads to $\pm 30\%, \pm 15\%$ and $\pm 3\%$ variation in the Young's modulus. As we have an uncorrelated field we can discretize the random field in the same way as the finite element mesh such that at every finite element we have a discretization point of the stochastic mesh. The value of the realization at the discretization point then determines the Young's modulus of one finite element.

Monte Carlo method

As the stiffness matrix of the SFEM system (2.27) contains a random variable the displacement, as the solution of the stochastic partial differential equation, is also a random variable. The Monte Carlo method is a very general and simple approach to gain properties of random variables of unknown distribution. The stochastic variables are replaced by realizations of their corresponding stochastic distribution. The problem renders then again deterministic and can be solved with a standard finite element computation. To account for the distribution of the stochastic input variables the realizations have to follow the input distribution and the simulation has to be done a large number of times, say N_{MC}. Mathematically, the law of large numbers lays the foundation that the simulations approximate the real solution. The realizations are gained by pseudorandom number generators, which are able to generate the needed distribution. The advantage of this sampling is that we are able to get an approximation of whole distribution of the solution variable. On the other hand, the large number of computations leads to a big numerical effort, which cannot be used for huge simulations. Moments of the response variable can exemplarily be calculated by their statistical equivalents. The first two moments, the expectation value and variance, for example can be computed by

$$\begin{aligned}\mathbb{E}(y(\boldsymbol{x})) &= \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}} y_j(\boldsymbol{x})\\ \mathbb{V}ar(y(\boldsymbol{x})) &= \frac{1}{N_{\mathrm{MC}}-1}\sum_{j=1}^{N_{\mathrm{MC}}}\left(y_j^2(\boldsymbol{x}) - N_{\mathrm{MC}}(\mathbb{E}(y(\boldsymbol{x})))^2\right)\end{aligned}$$

where y_j is a scalar variable of interest in the j-th simulation, for example a displacement, a strain or a stress value. Furthermore we are able to approximate a failure probability

by

$$P(y(\boldsymbol{x}) > y_{\text{crit}}) = \frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} 1_{[y_{\text{crit}},\infty)}(y_j(\boldsymbol{x})) \tag{3.6}$$

where $1_A(y)$ is the indicator function

$$1_A(y) = \begin{cases} 1, & y \in A \\ 0, & y \notin A \end{cases}$$

and y_{crit} is a critical value where failure occurs such that the sum in (3.6) counts the numbers of simulation where failure occurs. Summarizing, all essential parts of the SFEM are now determined and simulations can be executed.

3.3 Data-driven finite element method

To conduct data-driven simulations we have to define the data set which is used as it is an essential part of the simulation. Furthermore, we need to set up an iterative algorithm to get the optimal data points.

Data set simulation

The data set describes the material behavior in the data-driven framework and is the difference to the other methods. Therefore, it plays a crucial part in the upcoming simulation results. As we simulate several examples with different materials and aim for a more numerical comparison of the methods all data sets are synthetically gained by a known material law, except the ones gained from the RVE in Chapter 6. As the quality of the data-driven solution is also very dependent on the input data we need to clearly state what data sets are used and how this affects the results of the data-driven simulations. From an experimental point of view many methods are already available. Uniaxial or biaxial tension tests are generally feasible. The experimentally obtained data are typically collected in stress-strain or load-elongation diagrams, which can be converted to stress-strain data. Triaxial experiments are rather seldom. However, imaging-based technologies such as computer tomography scans, magnetic resonance imaging and digital image correlation are state-of-the-art methods to deduce the material behavior and can be used to generate such data sets [80, 101, 110]. Deducing the three-dimensional material behavior from the one dimensional is rarely possible but in [106, 107, 108] it is done for plastic and elastoplastic behavior.

While we state here how the data sets are simulated we refer to Section 4.2 for the interpretation and analysis of how the data set and our choice of sampling affects the

data-driven solutions. Two different types of data sets are used in simulations. The first one is without any random influences or in other words without stochastic noise. The other one is with random influences. When we talk about no random influences or no noise we mean that Hooke's law holds for every data point of the data set

$$\sigma^* = E\epsilon^* \qquad \forall\ (\epsilon^*, \sigma^*) \in \mathcal{D} \tag{3.7}$$

or respectively in three dimensions

$$\boldsymbol{\sigma}^* = \mathbb{C}\boldsymbol{\epsilon}^* \qquad \forall\ (\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) \in \mathcal{D}. \tag{3.8}$$

These data sets are used to investigate the behavior and underline properties of the data-driven method itself. Differences to the other finite element methods are not influenced or biased by random behavior of the data then. Data sets with stochastic noise are used when we want to simulate close to the reality data sets and when we want to compare the results of the given methods for uncertain material behavior. In general, measurements are affected by different errors, which may result from imperfections of the measured object, the measuring procedure, the instruments, environmental impacts, human failure and many more [43]. Therefore, instead of fulfilling the equations (3.7) and (3.8), measured data sets will be more likely looking as in Fig. 3.4.

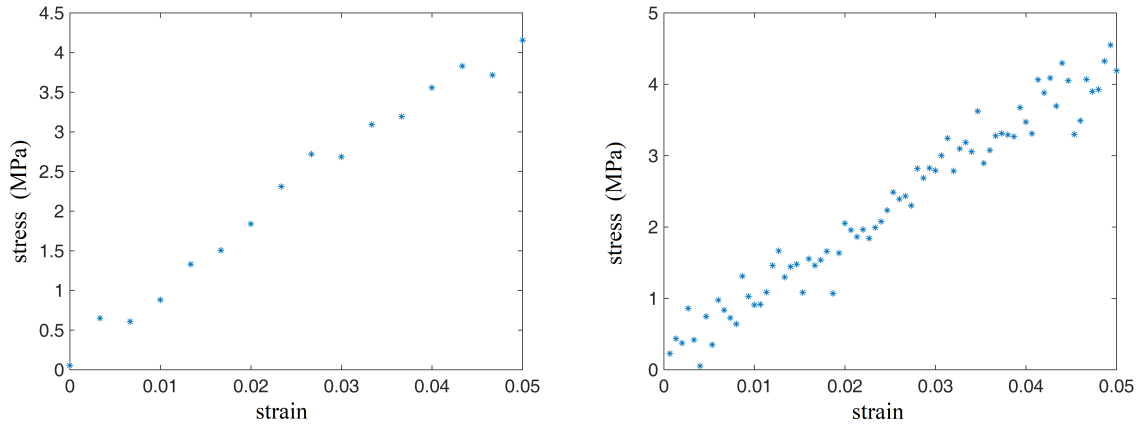


Figure 3.4: Exemplary uniaxial material data with 15 (left) and 75 (right) data pairs in tension. The measurement is afflicted by measuring errors and noise.

These errors may be divided into random and systematic ones. Random errors, which are also named statistical errors, result in scattering data and are caused by fluctuations of the measured quantity; they differ in their sign and absolute value. Random errors can be reduced by repeating the experiment. Systematic errors, in contrast, are deterministic and do not change if the experiment is repeated; they have the same absolute value and sign. An offset, which may be caused by insufficient calibrations, is a typical

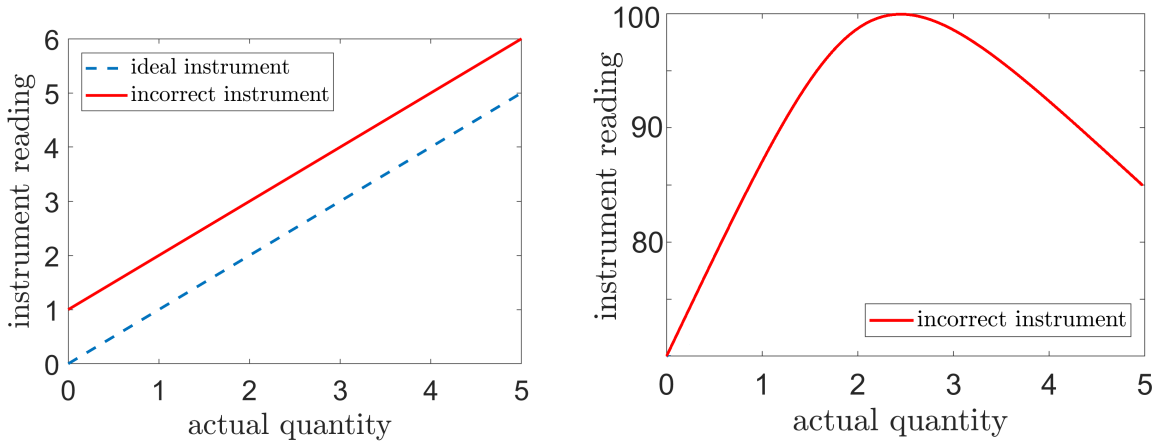


Figure 3.5: Examples for systematic errors: An offset results in an additive error (left) and the accuracy of a measuring instrument as a function of the magnitude of the measured quantity (right), which results in a multiplicative error.

systematic error. In the left image of Fig. 3.5 this error contribution is illustrated and also the accuracy of a typical load cell [44] on the right side which, like all measuring instruments, has only a certain loading range of optimal accuracy. Around the lower and the upper limit of this range the accuracy worsens, which results in a multiplicative error contribution. For the data set afflicted with stochastic noise and error we extend the linear equation by a multiplicative error $S_m(\epsilon)$, an additive error S_a and a stochastic variable $\mathcal{N}$

$$\sigma = S_m(\epsilon) E\epsilon + \mathcal{N} + S_a \,. \tag{3.9}$$

In this relation, $S_m(\epsilon)$ and S_a are functions describing the multiplicative and the additive part of the systematic error. Term N is the random error which is typically modeled by a normal distribution $\mathcal{N} \sim \mathcal{N}_{0,s^2}$ with mean 0 and variance s^2. Even though the random part is added it can inhabit a multiplicative part, if the mean and variance of the normal distribution depend on the strain as we have

$$\mathcal{N}_{\epsilon,\epsilon^2 s^2} = \epsilon \mathcal{N}_{1,s^2} \,.$$

In the corresponding multi-dimensional case the systematic errors, as well as the stochastic noise, have to be matrices

$$\sigma = \mathbf{S}_m(\epsilon)\mathbb{C}\epsilon + \boldsymbol{\mathcal{N}} + \mathbf{S}_a \,.$$

As we sample synthetic data we will assume no systematic errors. The simulation of both types of data sets are very similar then. The stochastic noise only needs to be

added in one step of the simulation.

One-dimensional data

Even though we want to conduct as many of the simulations in the three-dimensional setting as possible reducing to a one-dimensional problem often helps. At first we sample sets of data which can be generated by a uniaxial tension test and then generalize to more dimensions. We also want to briefly state that we talk about one-dimensional data because it describes a material which only takes up strains and stresses in one direction, even though a material data point $(\epsilon, \sigma) \in \mathbb{R}^2$ and the data set $\mathcal{D} \in \mathbb{R}^{n\times 2}$ will be of higher dimensions. If we assume ideal material behavior, the measured stress σ is a linear function of the applied strain $\sigma = E\epsilon$. Subsequently, it needs to be chosen which side is sampled. In this thesis, the stress space is always sampled by an equidistant grid. In one dimension the stress space is an interval $[-g, g]$; the number of data points then defines the distance between the data points and the data density. We will do simulations with different numbers of data points, which results in different data densities. It is obvious that with a greater number of data points a better result can be expected.

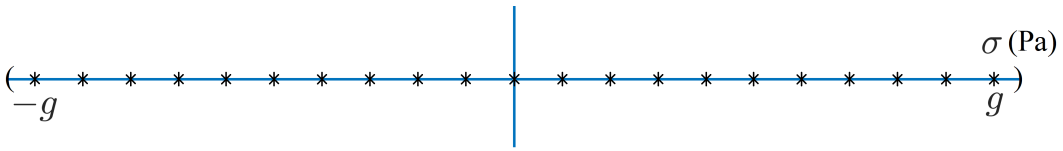


Figure 3.6: Sample interval of stress points. The distance between every data point is equal.

For the data set without a stochastic part the corresponding strain value is computed with a hidden Young's modulus, which we know for the data generation but is unknown in the finite element simulation. The set with stochastic noise is gained if we add a realization of the stochastic variable to the Young's modulus. We then gain a data set

$$\mathcal{D} = \{(\epsilon_i, \sigma_i)\}_{i=1}^{n} \in \mathbb{R}^{n\times 2}$$

of data points, which yields as an input for the finite element computation. For one specific data point we also write $z_i := (\epsilon_i, \sigma_i) = (\epsilon, \sigma)_i$.

Three-dimensional data sets

Further on, we consider three-dimensional computations of a material specimen. By now, these data can hardly be obtained from experiments but recent developments,

e.g. in-situ tomography scanning and also computational simulations of specimen with reconstructed microstructure, which allow to obtain the complete relation between the strain $\boldsymbol{\epsilon}$ and stress tensor $\boldsymbol{\sigma}$. In the linear elastic case we ideally have

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\epsilon} \tag{3.10}$$

We proceed as in the one dimensional case and sample an equidistant grid in the stress space. To reduce computational efforts in the data sampling as well as in the simulation we assume plane stress states in all three dimensional computations, if not stated otherwise. That means the stresses into the third direction vanish

$$\sigma_{33} = \sigma_{13} = \sigma_{23} = 0$$

and so it remains to cover the space of σ_{11}, σ_{22} and σ_{12}. Using the same limits for all three components we end up sampling the cube $[-g, g] \times [-g, g] \times [-g, g] = [-g, g]^3$. This time, the total number of data points is the number of points per axis to the power of three, which means that the curse of dimension increases the number of data points used really fast. For example, using 100 data points per axis leads to one million data points in the plane stress conditions. On the other hand, in full three dimensional simulations 100 data points per axis lead to one trillion($= 100^6 = 10^{12}$) data points, which adds a lot of numerical effort to the computation. The number of data points we use per axis will be between 10 and up to a maximum of 250. On every stress data point we can now apply a hidden compliance matrix $\mathbb{D} = \mathbb{C}^{-1}$ of the material we want to simulate. As a result we gain the noiseless data

$$\mathcal{D} = \left\{ \left(\boldsymbol{\epsilon}^i, \boldsymbol{\sigma}^i \right) \right\}_{i=1}^{n} \in \mathbb{R}^{n \times 12}$$

where we also write $(\boldsymbol{\epsilon}^i, \boldsymbol{\sigma}^i) = (\boldsymbol{\epsilon}, \boldsymbol{\sigma})_i =: \boldsymbol{z}_i$. To get the random inflicted data set we apply a random term, which only affects the Young's modulus but not the Poisson's ratio. Therefore, the random part reduces to a variable again. We can use the representation (3.5) again and apply a random part to the Young's modulus

$$\mathbb{C} = (1 + \mathcal{N})E \cdot \widehat{\mathbb{C}}\,.$$

With the stochastic afflicted compliance tensor the afflicted data set is generated. Note that for multi-dimensional data, a-priori assumed isotropy of the material can be used to increase data by axis rotations. This can be done by applying a rotation tensor $\boldsymbol{R} \in \mathrm{SO}(3)$ to each data point $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$ of the set. Stepwise rotations $(\boldsymbol{R}^T \boldsymbol{\epsilon} \boldsymbol{R}, \boldsymbol{R}^T \boldsymbol{\sigma} \boldsymbol{R})$ result in a multitude of new data points and reduces the effort of phase space sampling, cf. [67]. Here, however, we assume material uncertainties that can behave different in different directions. Therefore, we stick to the synthetic experimental data provided by the sampling. Kirchdoerfer and Ortiz [67] showed that assuming isotropy onto all data points can reduce the numerical effort as it is possible to minimize over the eigenvalues

of the material data first.

Data-driven algorithm

We derived the DD-FEM equations for the minimal distance in the previous chapter but we still have to account for the search of the optimal data points. In both systems of equations (2.39) and (2.40) the optimal data points are used but they are unknown prior. To find the optimal data points an iterative algorithm is used, which was also proposed by Kirchdoerfer and Ortiz [67]. The data-driven problem is a distance minimization problem between two sets in the strain-stress phase space, see Fig. 2.1 again. One is the material data set $\mathcal{D}$ that describes the material behavior and the other is a constraint set $\mathcal{C}$ which inhabits all strain-stress states, which fulfill the physical constraints onto the system. In both we are allowed to move freely to find the minimal distance.

Alternating projection method

For two convex sets $\mathcal{C}, \mathcal{D}$ the minimal distance problem

$$\min_{y\in\mathcal{C}} \min_{z\in\mathcal{D}} ||y - z|| \tag{3.11}$$

is a well-known problem, which is commonly solved by alternating projection methods [13, 14, 24]. An arbitrary starting point is chosen, say $y^{(0)} \in \mathcal{C}$, and the projection onto the second set $z^{(0)} = P_{\mathcal{D}}(y^{(0)})$ is computed. Then the projection back onto the first set is computed $y^{(1)} = P_{\mathcal{C}}(z^{(0)}) = P_{\mathcal{C}}(P_{\mathcal{D}}(y^{(0)}))$. Repeating this procedure we get two converging series, which lead to the minimizers $y^* \in \mathcal{C}$ and $z^* \in \mathcal{D}$, which define the minimal distance $||y^* - z^*||$ between both sets.
This procedure is now applied to the data-driven framework. For the sake of clarity we only denote here the global and already discretized version of the problem, which is solved in the end. In the appendix the corresponding local and continuous versions are also shown. Applying the procedure we can formulate the discretized data-driven problem as

$$\min_{\mathbf{y}\in\mathcal{C}} \min_{\mathbf{z}\in\mathcal{D}} ||\mathbf{y} - \mathbf{z}||_{\mathbb{C}^\circ} \tag{3.12}$$

where

$$\mathbf{y} = \left\{(\boldsymbol{\epsilon}, \boldsymbol{\sigma})_m\right\}_{m=1}^{n_m} \in \mathcal{C} \quad \text{and} \quad \mathbf{z} = \left\{(\boldsymbol{\epsilon}, \boldsymbol{\sigma})_m\right\}_{m=1}^{n_m} \in \mathcal{D}$$

are collections of data points at the material points out of the corresponding spaces. The global constraint set here is the Cartesian product of the local constraint sets $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_{n_m}$. The global data set is the Cartesian product of the local data sets that

are assigned to the material points $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_{n_m}$. The metric $|| \bullet ||_{\mathbb{C}^\circ}$ is the corresponding metric to the strain energy density and the numerical stiffness tensor $\mathbb{C}^\circ$

$$||\mathbf{z}|| = || \{(\boldsymbol{\epsilon}, \boldsymbol{\sigma})_m\} ||_{\mathbb{C}^\circ} = \left(\sum_{m=1}^{n_m} w_m \left(\boldsymbol{\epsilon}_m : \mathbb{C}^\circ : \boldsymbol{\epsilon}_m + \boldsymbol{\sigma}_m : (\mathbb{C}^\circ)^{-1} : \boldsymbol{\sigma}_m \right) \right)^{1/2} .$$

The optimal data points of the data-driven problem can be found as the argument of the prior problem

$$\arg \min_{\mathcal{D}} \left\{ \min_{\mathrm{y} \in \mathcal{C}} \min_{\mathrm{z} \in \mathcal{D}} ||\mathbf{y} - \mathbf{z}||_{\mathbb{C}^\circ} \right\} .$$

By $\mathbf{z}^{*(i)}$ we now denote the collection of data points, which is assigned to the material points of the finite element mesh in the i-th iteration. Starting with a collection of data points from the data set $\mathbf{z}^{*(0)}$ we can solve the sub-problem

$$\mathbf{y}^{(0)} = \arg \min_{\mathrm{y} \in \mathcal{C}} ||\mathbf{y} - \mathbf{z}^{*(0)}||_{\mathbb{C}^\circ} .$$

This step corresponds to the solving of the prior derived finite element equations and the conversion of the displacement and the Lagrangian into the strain-stress phase space by the kinematic equation

$$\boldsymbol{\epsilon} = \mathbf{B} \widehat{\boldsymbol{u}}$$

and equation (2.37)

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^* - \mathbb{C}^\circ \mathbf{B} \widehat{\boldsymbol{\lambda}} . \tag{3.13}$$

Kirchdoerfer and Ortiz proposed a random initialization of data points $\mathbf{z}^{*(0)}$ here, while in Leygue a self-consistent approach is used to get optimal starting points and a faster convergence rate. Then, with $\mathbf{y}^{(0)}$, the closest point in the data set can be derived by

$$\mathbf{z}^{*(1)} = \arg \min_{\mathrm{z} \in \mathcal{D}} ||\mathbf{y}^{(0)} - \mathbf{z}||_{\mathbb{C}^\circ} .$$

This problem is a simple nearest-neighbor problem where we search for the closest data points from $\mathcal{D}$ to given points $\mathbf{y}^{(0)}$. Once those data pairs are found for each material point one iterative step of the algorithm is completed. The loop continues with the solution of the linear equation system (2.39) and (2.40) again. Generalizing those two steps we can write for the k-th step of the algorithm

$$\mathbf{y}^{(k)} = \arg \min_{\mathrm{y} \in \mathcal{C}} ||\mathbf{y} - \mathbf{z}^{*(k)}||_{\mathbb{C}^\circ} \tag{3.14}$$

$$\mathbf{z}^{*(k+1)} = \arg \min_{\mathrm{z} \in \mathcal{D}} ||\mathbf{y}^{(k)} - \mathbf{z}||_{\mathbb{C}^\circ} . \tag{3.15}$$

In (3.15) we search for the data points, which are closest to satisfying the constraints, and in (3.14) the points of the constraint set closest to the data are searched. Both correspond to the respective projections of one set to the other. The algorithm stops when in two consecutive iterations all data points coincide. Alternatively, we can stop the algorithm when the decrement of the penalty function (2.31) is smaller than a certain threshold, for example $5\%, 1\%$ or 0.1%. This is a computing time-saver as one iteration can be numerically very expensive if big data sets are used. Only a small part of the precision is then given up for a decrease in computing time. As a solution of the data-driven method we will always use the latter representation. A flowchart of the DD-FEM procedure is given here, which summarizes the steps above, see also [66, 67].

i=0

1.) **Initialization of data points**
for m=1 ,2 ,... , n_m **do**
Choose $\left(\epsilon_m^{*(0)}, \sigma_m^{*(0)}\right)$ from $\mathcal{D}_e$
end;

i=1
2.) **Solving of the DD-FEM equations (2.39) and (2.40)**

Solve

$$\mathbf{K}\,\widehat{\boldsymbol{u}} = \boldsymbol{f}_u$$
$$\mathbf{K}\,\widehat{\boldsymbol{\lambda}} = \boldsymbol{f}_\lambda$$

for $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{\lambda}}$

3.) **Computation of trial strains and stresses**

for m=1 ,2 ,... , n_m **do**
$\boldsymbol{\epsilon}_m = \mathbf{B}_m^e \widehat{\boldsymbol{u}}^e$, $\boldsymbol{\sigma}_m = \boldsymbol{\sigma}_m^* + \mathbb{C}^{\circ}\mathbf{B}_m^e \widehat{\boldsymbol{\lambda}}_e$
end

4.) **New assignment of data points**

for m=1 ,2 ,... , n_m **do**
Search in $\mathcal{D}_e$ for the closest data point to $(\boldsymbol{\epsilon}_m, \boldsymbol{\sigma}_m)$ and assign it to $(\epsilon_m^{*(i)}, \sigma_m^{*(i)})$
end

5.) **Test for convergence**

if $|\frac{W^{\text{PY}}(i)-W^{\text{PY}}(i-1)}{W^{\text{PY}}(i)}| > c$ **then** i=i+1 and start reiteration at 2.)
else
end

The alert reader may have already noticed that we adopted the alternating projection method to the data-driven problem without saying something about the convexity of the sets $\mathcal{C}$ and $\mathcal{D}$. In fact, $\mathcal{D}$ as a discrete set is non-convex and it is possible that the proposed algorithm leads only to a local minimum. Finding a minimal distance between non-convex sets is a more challenging problem. A graphical illustration of both sets and why we still use the proposed algorithm is given in Fig. 3.7. Depending on the starting point the data-driven algorithm will lead to different solutions and distances. However, as seen in the image, those distances and solutions do not differ much from the global solution. Numerical experiments showed that the local minima are still close to the results of classical finite element computations [34, 67, 88, 89, 90]. Even more, if the data set is converging to a graph, the data-driven solutions converge to the classical solution [67]. With this in mind have a look especially at the examples in the beginning of the next chapter.

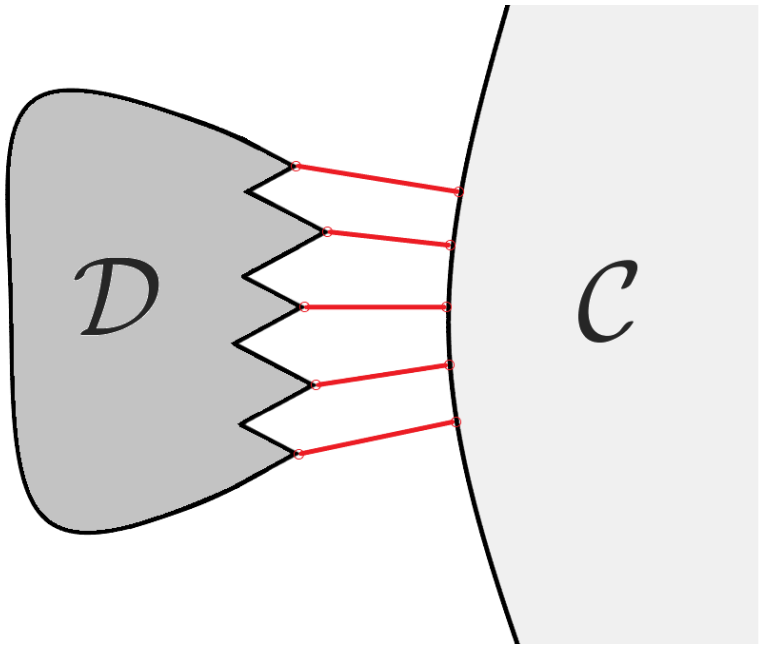


Figure 3.7: Illustration of the data and the constraint set. It is possible that the algorithm assigns a local minimum. However, they still approximate the mechanical state adequately for a dense data density.

3.4 Numerical examples

In the following, we want to briefly introduce three problems which are used to investigate the data-driven method in the next chapter. As seen before we will refer to the example

of a pulled rod. This one-dimensional example can be easily visualized and several properties of the data-driven framework will be explained on it. We just briefly recap the given formulas. A cantilever beam problem is introduced afterwards. This example is used for the studies towards the number of data points and the numerical stiffness tensor as well as for the comparison to the SFEM. A plate with a hole is used as a second example to verify that the results are independent from the choice of our problem. For both problems an analytic solution is at hand.

Tension rod

This basic example is meant to illustrate the data-driven method in a simple and visual way. In a one-dimensional problem the strain and stress are scalar values. Therefore, we can simply plot the material data in a stress-strain diagram and also explore the data-driven method visually. For given values of the length l, the cross-section A and the force F we can calculate the analytic solution

$$\sigma = \frac{F}{A}, \qquad \epsilon = \frac{\sigma}{E}.$$

Remember that there is no given Young's modulus in the data-driven problem. The material stiffness is inhabited in the data, which is used in the data-driven problem.

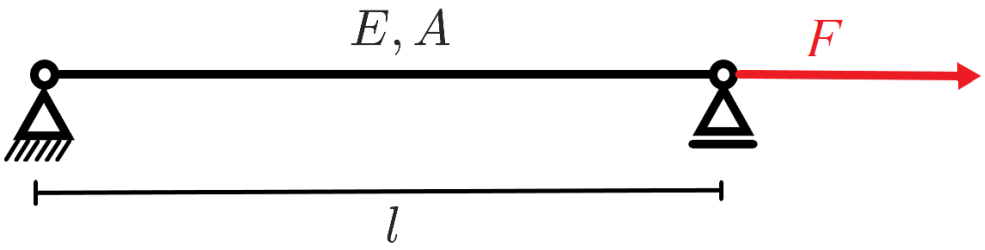


Figure 3.8: Geometry and boundary conditions of the tension rod. Remember that in the data-driven setting no Young's modulus is assigned.

Cantilever plate

The example on which we mainly investigate the properties of the data-driven method is a cantilever plate bending problem. A cantilever plate with a size of $2\,\mathrm{m} \times 0.5\,\mathrm{m} \times 0.1\,\mathrm{m}$ is loaded from atop, $p_0 = 4 \cdot 10^6 \mathrm{N/m}$. For the FEA the domain is discretized in $25 \times 10 \times 4$ linear brick elements with 8 integration points. More details are displayed in Fig. 3.9.

For comparison we refer to a classical FEA and to the analytical plane stress solution. Using Airy's stress function ansatz [45], we obtain from its partial derivatives the fol-

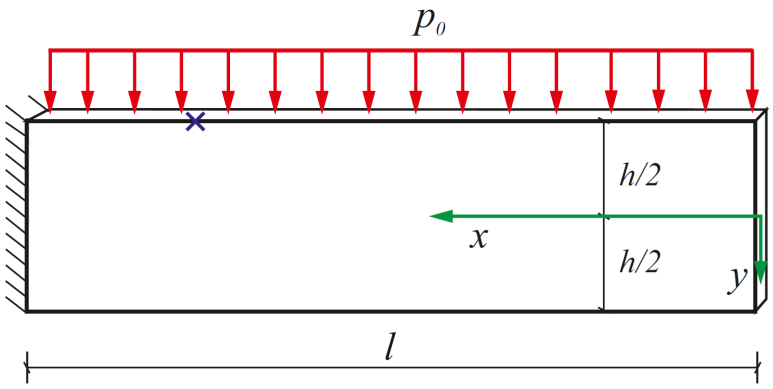


Figure 3.9: Geometry and boundary conditions of the cantilever plate.

lowing stress components:

$$\begin{aligned}\sigma_x &= \frac{p}{h^3}(6x^2y - 4y^3 + \frac{3}{5}h^2y),\\ \sigma_y &= \frac{p}{h^3}(2y^3 - \frac{1}{2}h^3 - \frac{3}{2}h^2y),\\ \sigma_{xy} &= -\frac{p}{h^3}(6xy^2 - \frac{3}{2}h^2x).\end{aligned} \tag{3.16}$$

An analytic deflection function cannot be derived in this way, so we restrict ourselves mainly to a comparison of the computed stresses.

Plate with a hole

A second investigative example is calculated to check if further effects, that do not occur with the cantilever plate, can be found. A plate with a circular hole in the center is studied. For symmetry reasons we reduce the problem to one quarter, i.e. a plate of $2\,\mathrm{m} \times 2\,\mathrm{m} \times 0.1\,\mathrm{m}$ with a hole of radius $r = 0.5\,\mathrm{m}$. Geometry and boundary conditions are displayed in Fig. 3.10. For the finite element simulation the quarter plate is meshed with 4611 linear hexahedra. In x-direction the plate is subjected to a tension of $p_0 = 10^7\,\mathrm{N/m^2}$. An analytic plane stress solution for the infinitely large plate exists, see e.g. [45]. It serves us as a reference because for $4a \leq b$ the maximum stress $\sigma_{x,\max}$ has an error of less than 6% [109].

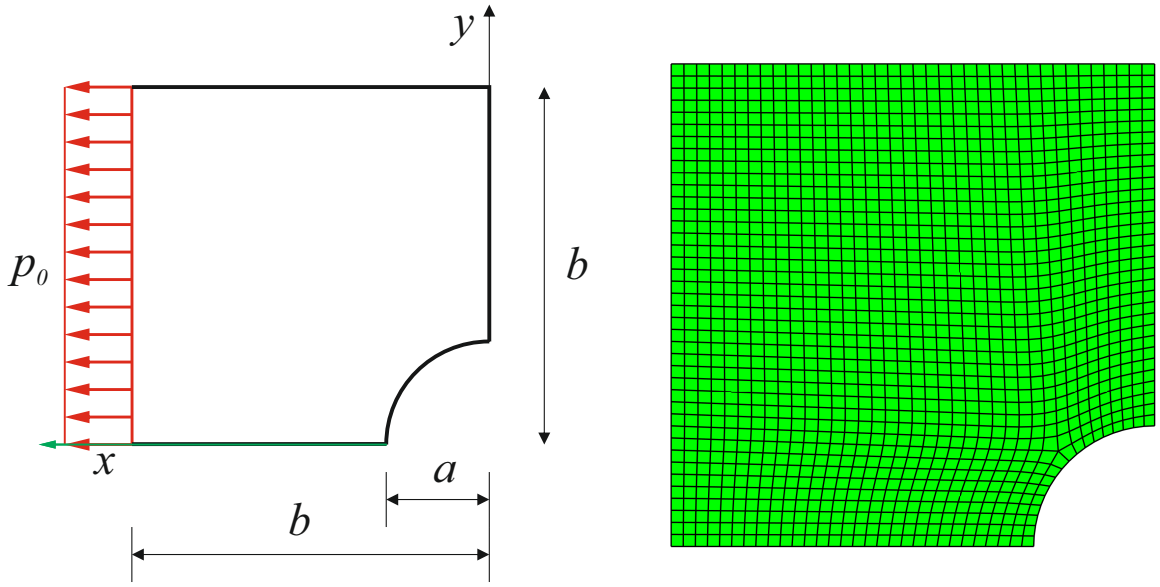


Figure 3.10: Plate with a hole: Geometry and boundary conditions on the left side, finite element mesh on the right side.

4 Data-driven, classical and stochastic FEM

In this chapter, the main part of the thesis starts where we begin to compare the data-driven approach to the other finite element methods. Before we compare the different approaches, we investigate the DD-FEM in a simple problem to present the properties of the data-driven algorithm in Section 4.1. First, a rod is considered where we first exemplarily solve the DD-FEM problem graphically. This is a straightforward task as strain and stress are one-dimensional and can be plotted easily. We then investigate four special cases which can occur in the DD-FEM setting: the incompleteness of the data set, outliers in the data, randomness in DD-FEM solutions and the use of multiple data sets. Afterward, we move on towards to the comparison between the DD-FEM and the classical FEM in Sections 4.2 and 4.3 depending on the two input variables of the DD-FEM, the data set and the numerical stiffness tensor $\mathbb{C}$. Both are investigated for the cantilever plate problem presented in 3.4. Then we progress to a comparison with the SFEM in Section 4.4. This chapter partially relies on the publications [71] and [72].

4.1 Basic properties of the data-driven finite element method

After we established the data-driven algorithm in the previous chapters we start with a simple calculation now. A rod is considered, which is described by one finite element. As the rod takes loads in only one direction, the strain and stress are scalar values. Then, the material describing data set is easy to plot and we can also derive the solution graphically.

A simple example of the data-driven algorithm

We consider a rod of $A = 1\,\mathrm{mm}^2$ cross-section, which is pulled by a force of $F = 6\,\mathrm{N}$. The constraint set is then defined by the constant line $\sigma = F/A = 6\,\mathrm{MPa}$. By adding a data set that displays the material behavior we can get the DD-FEM solution graphically in this easy case, see Fig. 4.1 . Starting the data-driven iteration at a random initialization $(\epsilon^{*(0)}, \sigma^{*(0)})$ we search for the closest point in the constraint set, this is $(\epsilon^{(0)}, \sigma^{(0)})$. Note

that, to deduce the shortest distance graphically here both axes have to be equally scaled, so the abscissa is multiplied by the Young's modulus and the numerical stiffness is set to the real Young's modulus $E^\circ = E$. This step later corresponds to the solving of the finite element system. From $(\epsilon^{(0)}, \sigma^{(0)})$ we search now for the closest data point from the data set, which leads to $(\epsilon^{*(1)}, \sigma^{*(1)})$. Redoing this four more iterations we get the solution of the data-driven method as $(\epsilon^{*(4)}, \sigma^{*(4)})$ coincides with $(\epsilon^{*(5)}, \sigma^{*(5)})$. The algorithm stops here as this data point is assigned two following iterations, which means that the data-driven method converged to its solution.

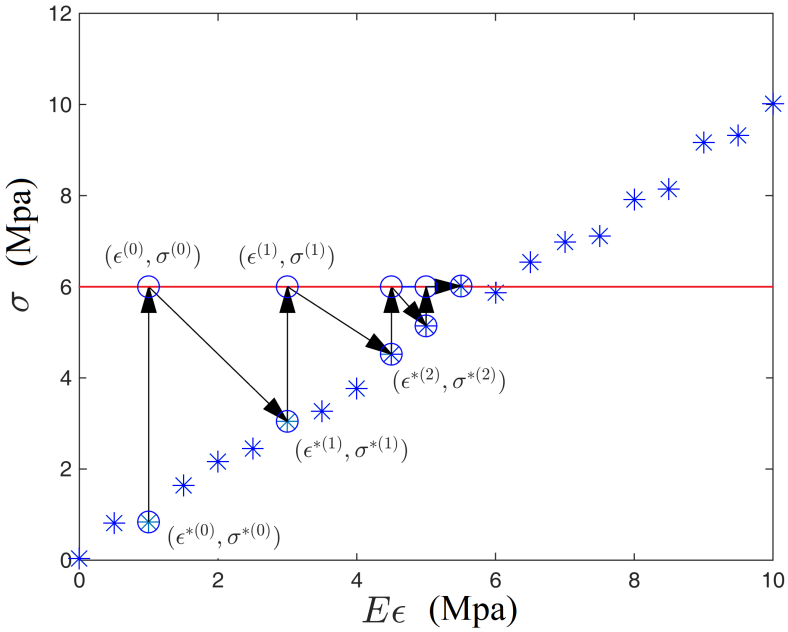


Figure 4.1: Iterations of the DD-FEM for a single element rod. From $(\epsilon^{*(0)}, \sigma^{*(0)})$ the DD-FEM iterates to $(\epsilon^{*(1)}, \sigma^{*(1)})$ via the constraint point $(\epsilon^{(0)}, \sigma^{(0)})$. Further on we iterate from $(\epsilon^{*(1)}, \sigma^{*(1)})$ to $(\epsilon^{*(2)}, \sigma^{*(2)})$ to $(\epsilon^{*(3)}, \sigma^{*(3)})$ to $(\epsilon^{*(4)}, \sigma^{*(4)})$ which is the solution of the DD-FEM. Labeling of the data points omitted later on due to readability.

This way of functioning of the data-driven methodology leads to some features that should not be forgotten in the later course of the simulations. In the following we will show these with simple examples.

Incompleteness of the data set

One should keep in mind that this method is not extrapolating. Using the same data set as before but a force of $F = 15\,\mathrm{N}$ leads to a significant distance between the data and

the constraint set, see Fig. 4.2. The data-driven method still yields the highest stress value in the data but underestimates the real solution. One should be aware that the data needs to cover the regions of the phase space of stresses and strains, which can occur in a particular problem. Obviously, stresses and strains are unknown a-priori such that estimates need to be gained by expert knowledge for example. However, most of the experimental setups can load materials until their failure such that enough data can be recorded. Furthermore, the method can indicate such behavior when the maximal strain or stress data points are often assigned in problems that inhabit more than one element.

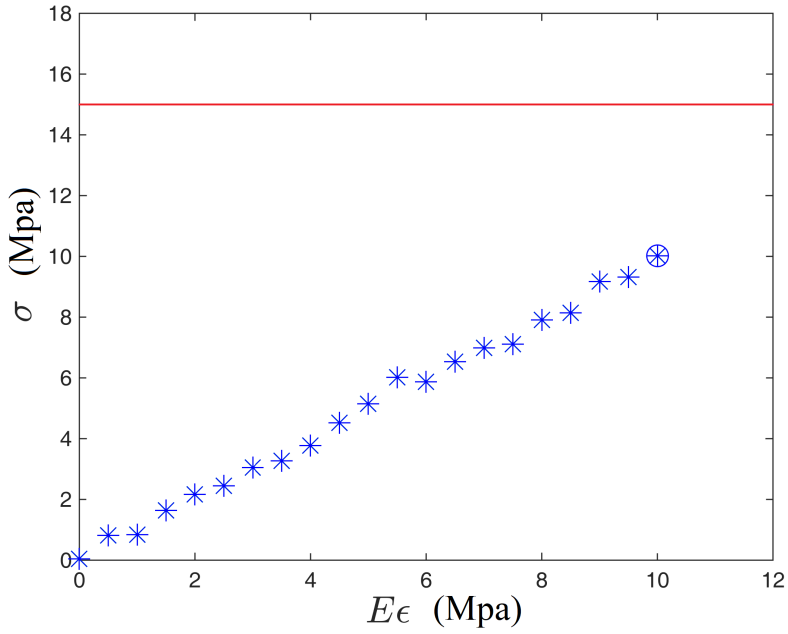


Figure 4.2: The data set covers not enough of the phase space. The data driven solution (circle) is not reasonable.

Outlier in the data

As already described, measured data sets can always be afflicted with errors. In the next data set, see Fig. 4.3, there is an obvious outlier in the data set. As we see in the iteration of the data-driven algorithm, the outlier dominates the solution of the data-driven method significantly. Kirchdoerfer and Ortiz proposed in [68] a maximal entropy version, called max-ent data-driven algorithm, which is robust towards outliers. They use cluster analysis to assign a weight to every data point that renders its relevance inside of the data set. A sum of the weighted data points is then used instead of a single data point. The distance minimizing algorithm is regained for the limit of the

temperature towards infinity. We need to note that all simulations in this thesis use the distance minimizing algorithm. The idea in this thesis, to use the data points themselves to describe the material uncertainty, leads to the desire to use the data as untreated and raw as possible. Therefore, the distance minimizing algorithm is used as the max-ent algorithm uses weighted sums of the data points and smears the data points. The assumption, which needs to be made then, is that significant outliers, which are consequences of human failure or measuring failure are removed by expert knowledge in the pre-processing step. All data sets used in this thesis are gained by adding a stochastic noise with a small variance to a hidden Hooke's law and therefore fulfill this assumption.

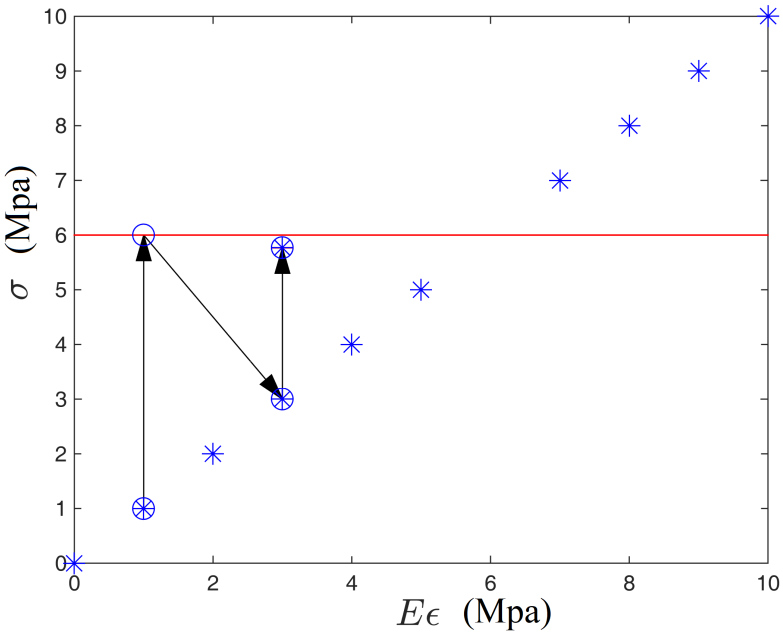


Figure 4.3: The data set inhabits an outlier that disturbs the results of the data-driven algorithm significantly. The outlier is too close to the constraint set for the others to play a role.

Random initialization points

One important point to understand is that even though we say that the data set includes material uncertainty, the DD-FEM is a deterministic method. A-priori, there is no random part in the method provided that the starting points are not chosen randomly. Randomly choosing starting points already adds a stochastic part to the method but not in the classical sense that it is in the material variable. However, using random starting points already leads to non-unique solutions as described in Section 3.3. The reason for

this is shown in Fig. 4.4. Two different starting points of the algorithm are chosen here, the points $(1, 1)$ and $(9, 9)$. Thinking of the double minimum iteration, the steps of the data-driven iteration can be found easily. In the end, we have two different solutions as they are fixed points of the given problem. The projection onto the constraint set of the point (5,5) leads to the point (5,6). However, the closest data point is then again (5,5). The same happens for the other starting point at the point (7,7) the data algorithm stops and yields it as a solution. Obviously, this data set is constructed that close to the intersection of a virtual function of the data set and the constraint set no data point is given but we have to keep in mind that this happens all the time just on different scales. Exemplarily, having every strain increase of 0.1 a data point except $\epsilon = 6$ leads to the two solutions 5.9 and 6.1. Increasing the precision again by a magnitude we gain 5.99 and 6.01 decreasing the error significantly. This emphasizes the need for a good phase space sampling and also shows how the convergence shown in [67] works if the data approximates a graph. Furthermore, we see that a starting point closer to the constraint set needs fewer iterations than other starting points, see also [78].

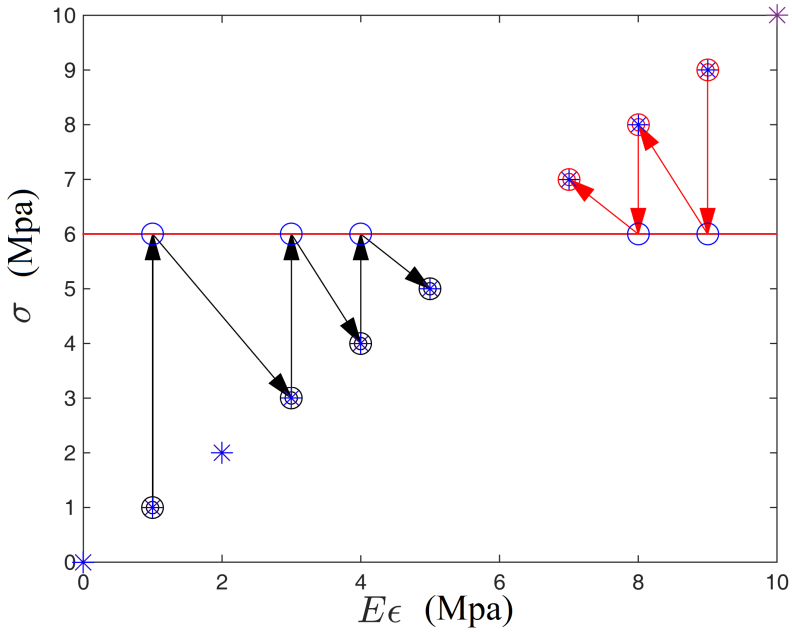


Figure 4.4: Data-driven iteration for two different starting points ending up in different solution points.

Multiple data sets

In the last example of this section we look towards using multiple data sets in one simulation. Typically, experiments are repeated several times to verify prior results.

With different realizations of experiments one can gather several data sets, this means we have data sets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, ...$ of repetitions of a tension test for example. This leads to the situation in Fig. 4.5: at given strain points we have measured different values of stress. In Fig. 4.5 there are exemplary three different data sets of the same material. However, a small difference is measured in the elastic modulus. In the data-driven simulation of Fig. 4.5 the data sets were all combined, so $\mathcal{D} = \cup_i \mathcal{D}_i$, and put into the data-driven algorithm. The data-driven algorithm now only assigns data points that are closer to the constraint set and the method stays at the 'outer' measured data points. Several simulations with distinct data sets could be preferential and this is where we will connect the data-driven algorithm to the stochastic finite element method.

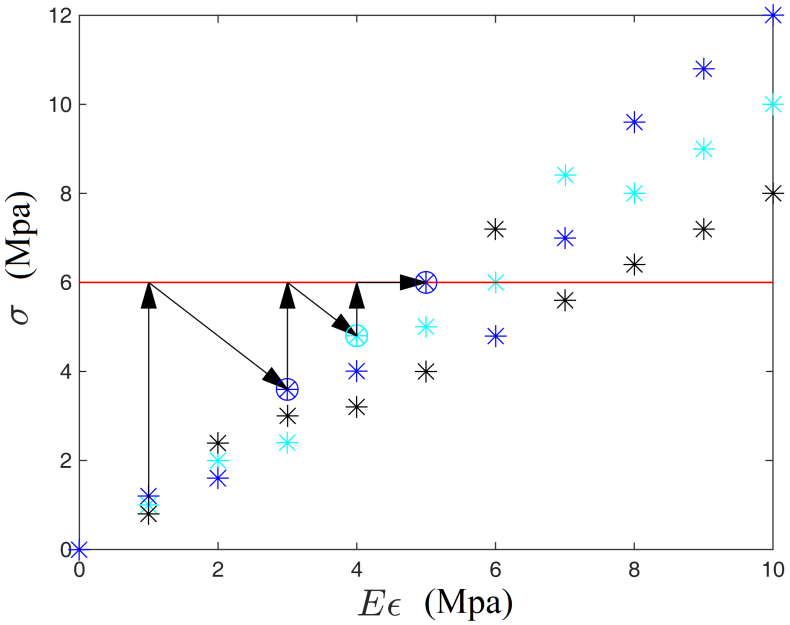


Figure 4.5: Data-driven iteration for multiple unified data sets.

Concluding the prior examples, the data set should meet the following requirements:

- the data set should cover the whole phase space of strains and stresses that will occur in the simulation,
- the data density should be as high as possible, or other put the distance between the data points should be as low as possible,
- unresonable outliers need to be removed prior to the simulation

While the first and third requirements will always be fulfilled in our generated data sets, the data density will be the focus of investigation in the next section. The way the data sets are sampled in this thesis is very conservative. As already explained a cubic

grid in the stress space is used. Most of the time, a certain number of data points is predefined. The size of the cube then defines the data density of the data set. The chosen size of the cube needs to be above the maximal occurring stress. To stay as close to reality as possible, we acted like the maximal stress is unknown and used conservative sizes of the stress grid. Furthermore, in most problems a certain stress coordinate will peak out while another may be very small. In a cube the data density is the same for all components and the cube size is orientated towards the maximal stress values. Therefore, the data density can be insufficient for the smaller stress component.

4.2 Results with respect to the data size

In their seminal work, Kirchdoerfer and Ortiz [67] laid their focus more towards convergence studies than on the result of a data-driven computation itself. The results shown here are also part of the paper [72], which showed detailed results of a data-driven computation first. A comparison of the classical finite element solution and the analytic solution is given. These results are combined with a study towards the number of data points where the obvious is shown: More data means more information and better results. The data set describes the material behavior and therefore plays a crucial role in the result of the simulation. The two properties that characterize the data set are the data density, say how many data points are in a certain area, and the fluctuations of the data. The fluctuations mainly describe the randomness of the material and are investigated in Section 4.4.

We start presenting the solutions with the previously introduced cantilever plate. In Fig. 4.6 the stress distribution of σ_x is computed with the DD-FEM for different sizes of the data set above the classical FEM solution. The sizes of the data set are $11^3, 31^3, 81^3$ as we are in the plane stress state. The cubic subspace $[-2{\cdot}10^8, 2{\cdot}10^8]^3$ of the stress space is sampled in each case such that the data density of data points per space increases. The data set represents the material data base of a ZAMAK-5 alloy, which corresponds to a Young's modulus $E = 85\,\mathrm{GPa}$ and Poisson's ratio $\nu = 0.3$ [63]. It is clear that for coarse data sets the solution cannot be as smooth as in the classical FEA because the choice of (ϵ^*, σ^*) pairs is very limited. Here, we see how the stress distribution evolves from the coarse data set to the more dense data sets. For the coarse $\mathcal{D}(11^3)$ data set the stress distribution is rather patchy. In contrast, the results for the $\mathcal{D}(151^3)$ data points show a smooth transition of the stresses and agree well with the reference solution as the jumps between the data points are small. In between the results improve for every increase in the number of available data. This observation corresponds to the results of Kirchdoerfer and Ortiz [67]. They showed convergence of the data-driven model to the elastic solution for a sequence of data sets $\mathcal{D}_k$ that approximate increasingly closer to a material law. Until here, the comparison was only visually. Now we introduce a quantitative error to measure the error of the DD-FEM in contrast to the classical

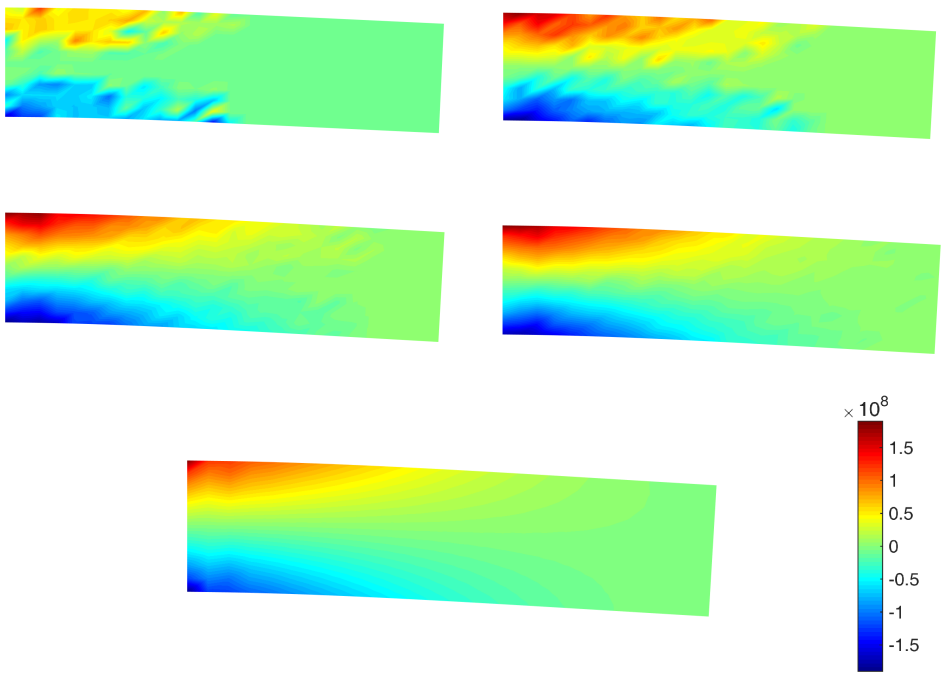


Figure 4.6: Data-driven computations of the normal stress σ_x in the cantilever plate: solutions obtained with 11^3 (top left), 31^3 (top right), 81^3 (middle left), 151^3 (middle right) data points and reference FEM solution (bottom). Displacement magnified for visibility reasons.

solution. To do so, the elastic energy density (2.8) is employed to define a root mean square (RMS) distance for the strains

$$\boldsymbol{\epsilon}_{RMS} = \left(\frac{\int_\Omega \Psi^e(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\text{ref}}) \, \mathrm{d}\Omega}{\int_\Omega \Psi^e(\boldsymbol{\epsilon}_{\text{ref}}) \, \mathrm{d}\Omega} \right)^{1/2} \tag{4.1}$$

and to evaluate the stresses its dual (2.21) is used

$$\boldsymbol{\sigma}_{RMS} = \left(\frac{\int_\Omega \Psi^*(\boldsymbol{\sigma} - \boldsymbol{\sigma}_{\text{ref}}) \, \mathrm{d}\Omega}{\int_\Omega \Psi^*(\boldsymbol{\sigma}_{\text{ref}}) \, \mathrm{d}\Omega} \right)^{1/2} \tag{4.2}$$

where the reference solution is provided by the classical FEM. In Table 4.1 the distance values are listed for the data-driven solution with different sizes of data sets. The strain and stress distance coincide here as we use a data set with no uncertainties and the correct choice of $\mathbb{C}^\circ$. The values confirm what the graphical comparison showed: the distance decreases when the number of data points increases or, in other words, when the distance between the data points reduces. The DD-FEM solver is then able to

find data points closer to the 'exact' solution. Additionally, the global penalty function (2.31) is evaluated. The rather big values of the global penalty function should not confuse as the function is not normalized and depends on the magnitude of strains and stresses. The value labeled here as data distance is the minimal Euclidean distance between the stress values of two arbitrary data points of the data set. This value defines the maximal accuracy and decreases for an increasing number of data points obviously. The maximum displacements $u_{\max}$ are close to each other. An interesting fact is that the data-driven method with minimal data approximates the displacement here still well. The reference finite element solution is about 2% softer, its maximal displacement is 9.46 mm. Even though the maximal displacement is close to the classical reference solution and the visual comparison also showed a good coincidence the distance value of 11% may surprise a little bit. The reason here for this relatively high error are two reasons. First, about 33% of the absolute distance results from ϵ and σ components related to the third direction. Exemplary calculations showed that using the full set of tensor components solves this problem but an actual computation with 151^6 instead of 151^3 data points is rather costly. We address the problem of large data sets later in Chapter 5.

data set	$\boldsymbol{\epsilon}_{RMS}$	$\boldsymbol{\sigma}_{RMS}$	$W^{\text{py}}\,[J]$	data distance [MPa]	$u_{\max}\,[mm]$
$\mathcal{D}(11)$	0.5551	0.5551	$2.18 \cdot 10^8$	40	9.18
$\mathcal{D}(31)$	0.3241	0.3241	$2.64 \cdot 10^7$	13.3333	9.14
$\mathcal{D}(81)$	0.1508	0.1508	$5.70 \cdot 10^6$	5	9.24
$\mathcal{D}(151)$	0.1147	0.1147	$3.29 \cdot 10^6$	2.6667	9.27

Table 4.1: Distances of the data-driven simulations for the cantilever plate problem.

Second, the lateral stress σ_y and the shear stress σ_{xy} are by orders of magnitude smaller than σ_x. It reveals a problem of the DD-FEM. The σ_y and σ_{xy} values are so small that the data-driven approach with the given data is unable to reproduce them adequately. We compute zero stresses because the $\{0, 0\}$ pair is closer to the optimal solution than any other ϵ-σ pair. In order to map all components adequately, a very dense data set would be required. This, however, would raise the computational effort again. We want to remark here that the data is purposely generated like that by ourselves. We are obviously able to adjust the data set for the smaller values of the shear and lateral stresses. The sampling of a cube-like subspace is not mandatory at all. Using adjusted intervals and densities for the different components is possible here but it may pervert the truth. In experiments we expect the measurement accuracy to be the same for all directions. The data set is chosen here to compute the biggest stress component adequate. We, therefore, see no reason here for an extra adjustment of the data in the other directions even though it yields better results regarding the root mean square distances.

In Figures 4.7 and 4.8 the normal stress components of the classical finite element computation and the data-driven approach are compared to the analytic solution. In Figure 4.7 it can be seen that there is no big difference between the finite element methods to the analytic solution. This is also confirmed by the paths shown in Fig. 4.8 where something else needs to be highlighted. The data-driven solution only takes up values of the data set such that the path can not be smooth. Some kinks can be seen in the plot of the data-driven solution.

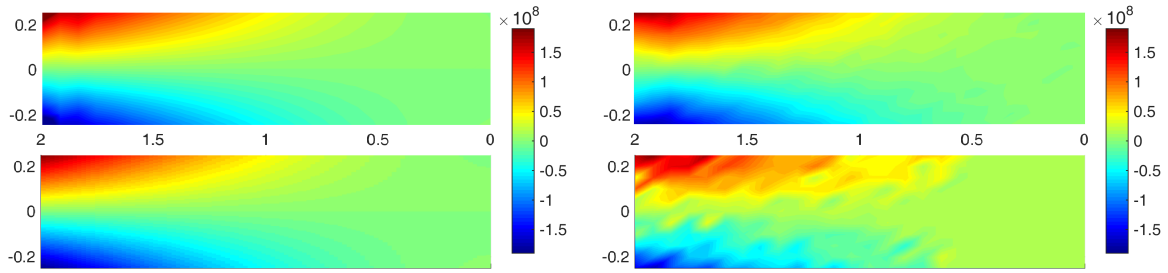


Figure 4.7: Comparison of analytical and computed stress distribution. Left: normal stress σ_x from standard FEA and analytical solution (3.16), Right: DD-FEM for 151^3 and 31^3 data pairs.

Another small aspect is the mapping of the singularity at the clamp. From linear elasticity we know that the stresses computed in a standard FEA are arbitrarily high here. This effect can not happen in the data-driven setting. The maximal value that can be computed here is the maximal value in the data. However, for the given mesh here the values of the finite element method are close the analytic solution of $\sigma_x \approx \pm 1.91 \cdot 10^8$ Pa.

For further insight we also conducted a second example here. The plate with a hole introduced in Section 3.4 is also computed with different sizes of the data set. The results of the computation can be seen in Table 4.2. We see the obvious increase in precision for larger data sets again. The displacement of the classical finite element solution is 2.69 mm and is also approximated very well by all but the coarsest data set here. Furthermore, the root mean square distances to the classical solution is clearly smaller than in the cantilever plate problem. While this can be attributed to a small part to a lesser error in the third direction this comes mainly from that the data just fits the stress and strain distribution better here.

In Fig. 4.9 we also see that the finite element computations do not reproduce the analytic solution perfectly. However, the maximal values of displacement and stress are close to each other. In the data-driven solution we see the kinks again due to the discrete nature of the data set.

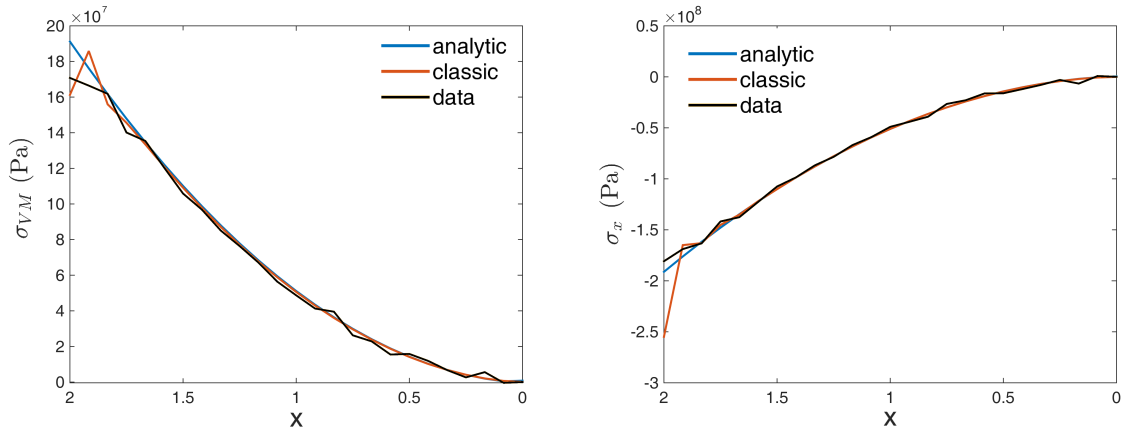


Figure 4.8: Stresses along the horizontal boundary of the cantilever plate, $y = \pm 0.25$m, $z = 0$, for the analytical solution, the classical FEM and the DD-FEM; von Mises stress along the top boundary (left) and σ_x along the bottom (right).

data set	$\boldsymbol{\epsilon}_{RMS}$	$\boldsymbol{\sigma}_{RMS}$	$W^{\text{py}}\,[J]$	data distance [MPa]	$u_{\max}\,[mm]$
$\mathcal{D}(11)$	0.3167	0.3167	$2.11 \cdot 10^8$	80	2.19
$\mathcal{D}(31)$	0.1404	0.1404	$3.86 \cdot 10^7$	26.6667	2.59
$\mathcal{D}(81)$	0.0560	0.0560	$7.60 \cdot 10^6$	10	2.67
$\mathcal{D}(151)$	0.0215	0.0215	$2.67 \cdot 10^6$	5.3333	2.69

Table 4.2: Results of the data-driven simulations for the plate with a hole problem.

4.3 Results with respect to the numerical stiffness tensor $\mathbb{C}^\circ$

Besides the data, the numerical stiffness tensor $\mathbb{C}^\circ$ is the second input, which is only needed by the DD-FEM. While the data is measured and does not need to be chosen, values for numerical stiffness tensor need to be estimated. To work as an adequate weight in (2.30), it should be as close as possible to the real material behavior. It is possible to get estimators from the data set, which, however, will still be off some percent. In the following, we present a parameter study, which assumes a small error in the choice of $\mathbb{C}^\circ$. Namely, we assume that the choice of ν is correct and we have a percentage error in the elastic modulus such that

$$\mathbb{C}^\circ = \alpha \cdot E \cdot \widehat{\mathbb{C}} \tag{4.3}$$

where $\widehat{\mathbb{C}} = 1/E \cdot \mathbb{C}$. For a value of $\alpha = 1$ the numerical stiffness coincides with the real hidden stiffness. While values below one correspond to a lower stiffness and values

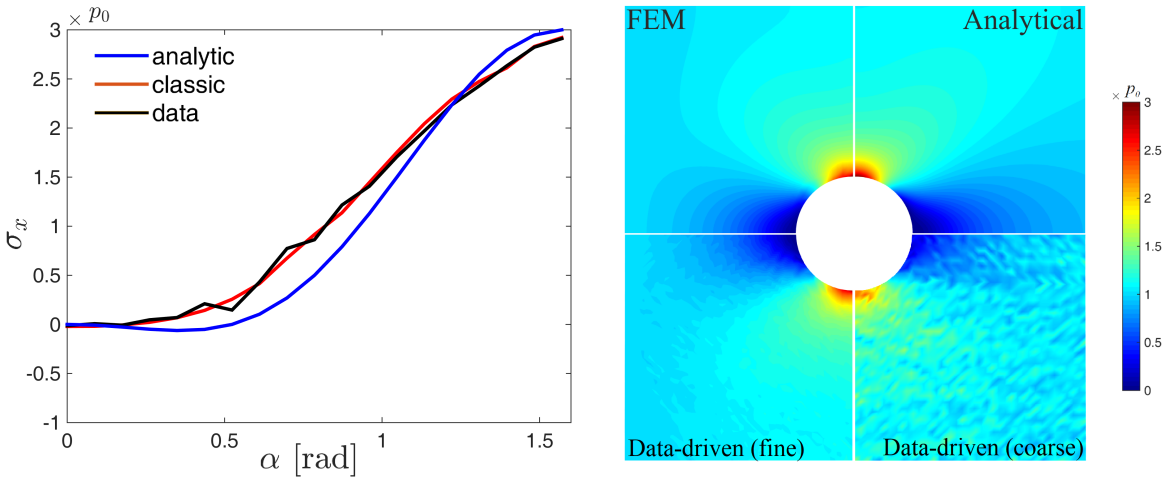


Figure 4.9: Comparison of the stress σ_x along the hole for the classical and data-driven FEA with an analytic solution (left). A visual comparison is shown on the right with two different data-driven computations (fine with $\mathcal{D}(151)$, coarse with $\mathcal{D}(51)$.

above one to a higher stiffness. The parameter study is carried out on the cantilever plate problem from the prior section with 100^3 data points. The whole results of the parameter study are shown in Table 4.3. For every value of α the value of the global penalty function W^{py}, the number of data iterations, the root mean square error of the strains and the maximum displacement are shown. In Fig. 4.10 the results are also displayed visually.

Going into detail, the DD-FEM shows an interesting and robust behavior against errors in the numerical stiffness tensor. In the first two images of Fig. 4.10, we see that the values of the global penalty function and the root mean square error is the lowest for the correct numerical stiffness. For the root mean square error, this is an expected result because this is a comparison of the data-driven solution and the reference solution, which is unknown in a real scenario. The global penalty function is a function of the difference of the computed trial stresses and strains and the data points multiplied with the numerical stiffness. Regardless of whether the stiffness is estimated too low or too high, it also is the lowest for the correct number of the numerical stiffness. This behavior also enables a simple approach for better estimates. Comparing two simulations with different numerical stiffness values, a next approximation step can be done towards the direction of the decreasing global penalty value. The same behavior is also seen for the number of data iterations. For bad choices of the numerical stiffness, the number of data iterations increases and good values have the lowest number of data iterations. The discrete values, however, do not enable small adjustments in the choice of the numerical stiffness. The displacement shows the robustness of the DD-FEM against bad choices of

α	W^{py}	no. of iterations	RMS_ϵ	max. displacement
0.5	$2.447 \cdot 10^7$	15	0.1683	$9.421 \cdot 10^{-3}$
0.75	$5.415 \cdot 10^6$	12	0.1683	$9.397 \cdot 10^{-3}$
0.9	$4.699 \cdot 10^6$	10	0.1434	$9.370 \cdot 10^{-3}$
0.95	$4.579 \cdot 10^6$	9	0.1381	$9.334 \cdot 10^{-3}$
1	$4.470 \cdot 10^6$	9	0.1337	$9.309 \cdot 10^{-3}$
1.05	$4.526 \cdot 10^6$	9	0.1345	$9.265 \cdot 10^{-3}$
1.25	$4.722 \cdot 10^6$	10	0.1365	$9.214 \cdot 10^{-3}$
1.5	$5.213 \cdot 10^6$	10	0.1415	$9.041 \cdot 10^{-3}$
1.75	$6.205 \cdot 10^6$	12	0.1612	$8.888 \cdot 10^{-3}$
2	$7.676 \cdot 10^6$	14	0.1937	$8.569 \cdot 10^{-3}$

Table 4.3: Results of the parameter study for different inputs of the numerical stiffness tensor $\mathbb{C}^\circ = \alpha E \widehat{\mathbb{C}}$.

the numerical stiffness. Although the displacement shows the expected behavior that for an increasing stiffness we get a decreasing displacement but it is very small compared for the change in α. As the numerical stiffness does not change the data itself and the material behavior the results differ only a bit. With a value of $\alpha = 2$ the displacement in the classic finite element analysis would be half of the prior results.

4.4 Comparison of the data-driven and stochastic finite element method

Next, we want to compare the ability of the data-driven method to include uncertain material behavior. Therefore, we compare it to the stochastic finite element method. The advantage that is hoped for is that, just as with the material description, a model-free description can be achieved. One problem of stochastic and uncertainty approaches is the description of the uncertain variable. Experiments are often done more than once to confirm results and determine a variation. However, often it is hard to conclude a distribution with their parameter from a small number of experiments. The data-driven approach got the advantage that no material defining variables are needed, the data is just put in. The idea is visualized in Fig. 4.11. The data describes the mean functional relationship as well as the variance around it.

When multiple data sets are at hand, at one measuring point several data points are

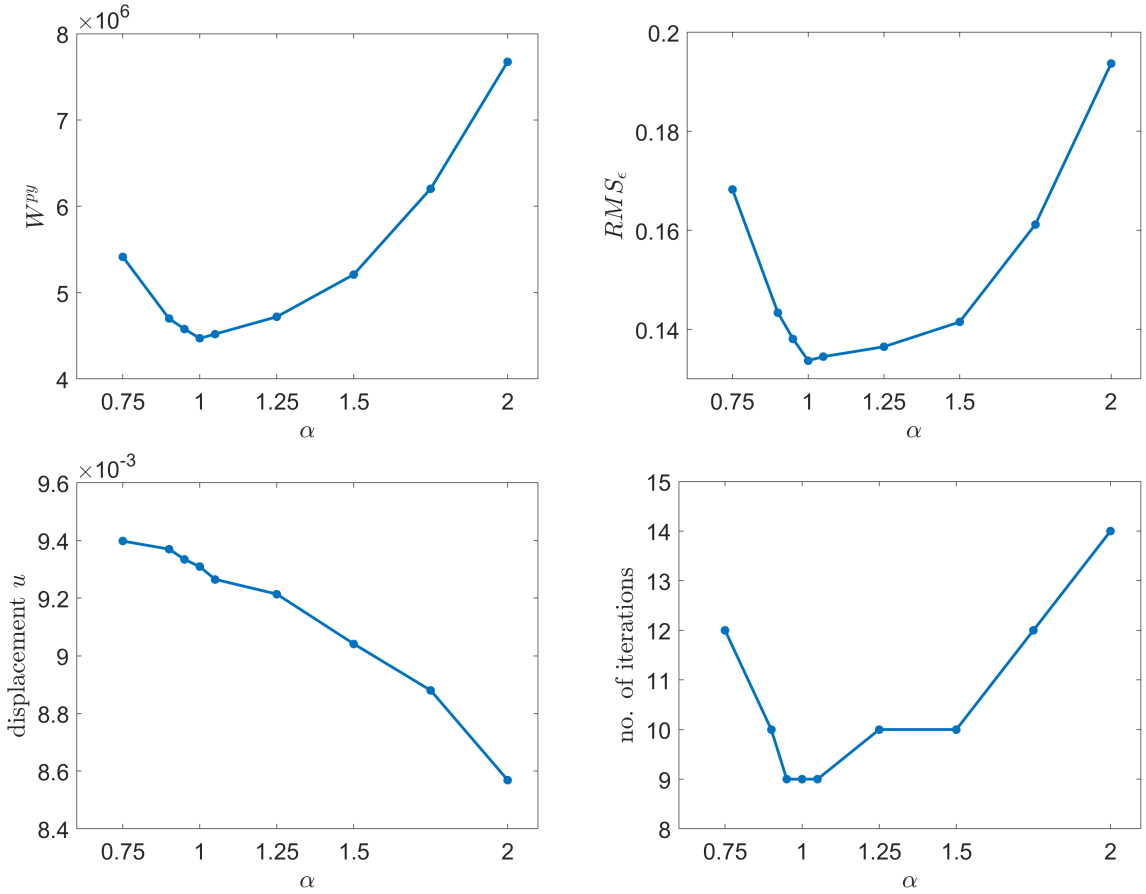


Figure 4.10: Global penalty function, root mean square error, displacement and the number of iterations plotted against the input value α.

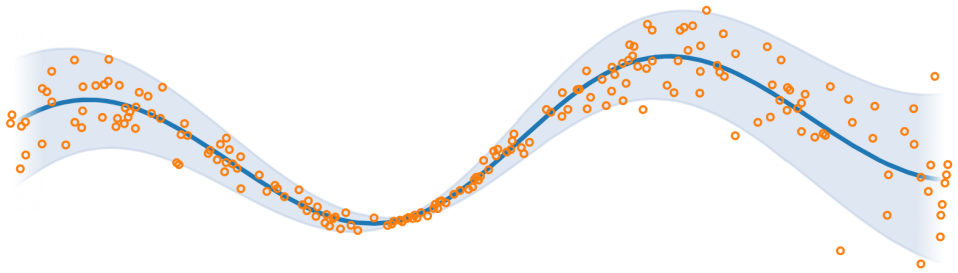


Figure 4.11: The idea of the uncertainty consideration for the data-driven framework: The data describes a given functional relationship and its variation well, from [3].

gained, which describe the material behavior with its small differences. With multiple data sets there are several possibilities to handle them:

- combine the data sets to one big set, which includes all data points
- data-driven computations with one data set for all elements
- data-driven computations with a random data set for every element

The first option is only listed here for the sake of completeness. It generates a data set that gathers all information in one data set but it prevents the uncertainty considerations. It also leads to the behavior of the last example shown in Section 4.1 at the beginning of the chapter. In the case of the second and third option we can compute solutions for every combination of data sets. The second case describes the situation in which the stiffness of the whole model depends on one variable. The third case represents the possibilities of a stochastic random field with local different material behavior. Histograms can be analyzed in both cases afterwards and we will investigate both of them therefore.
We now assume we have n noisy data sets $\mathcal{D}_1(d_1), \ldots, \mathcal{D}_n(d_n)$ where $d_1, d_2, \ldots, d_n$ are the number of data points included. Those do not have to be the same necessarily but due to the easier generation we set $d_1 = d_2 = \ldots = 71^3$. We do not use the highest amount of data points here as many simulations have to be conducted here. Numerical cost is saved therefore. The data sets are generated as explained in Section 3.3 with given variances $s^2 \in \{0.1^2, 0.05^2, 0.01^2\}$. For comparison reasons the Gaussian noise is simulated with the same variances. Inside a data set the noise is produced by Gaussian variable with mean 0 and variance 0.01^2. The quantity of interest is the maximal displacement of the cantilever plate and then for the plate with a hole. The results are only displayed for the highest variance as the results do not differ at all except the values of displacement and variation. We also want to outline that we do investigate the variation of the solution due to the different data sets. The variation is not affected by the random initialization of the data points. This effect is reduced by the data density as explained in the third example in Section 4.1.

One set for all elements

Simulations were conducted with $n = 50, 250, 500, 1000$ data sets used. In Fig. 4.12 we see the results for the different sizes of elements. With more data sets at hand we get more information about the distribution of the displacement. Just as a reminder: the maximal displacement of the deterministic solution is $9.42 \cdot 10^{-3}$ mm which is close to the center of every distribution here. However, it seems unrealistic to have 500 data sets. Nevertheless, statistics of the distribution, like the mean, the variance or minimal and maximal values, are already more or less recognizable in the upper two images with lesser data sets. In the last image the distribution of the DD-FEM with 1000 data sets is compared to a stochastic computation. The corresponding SFEM computation

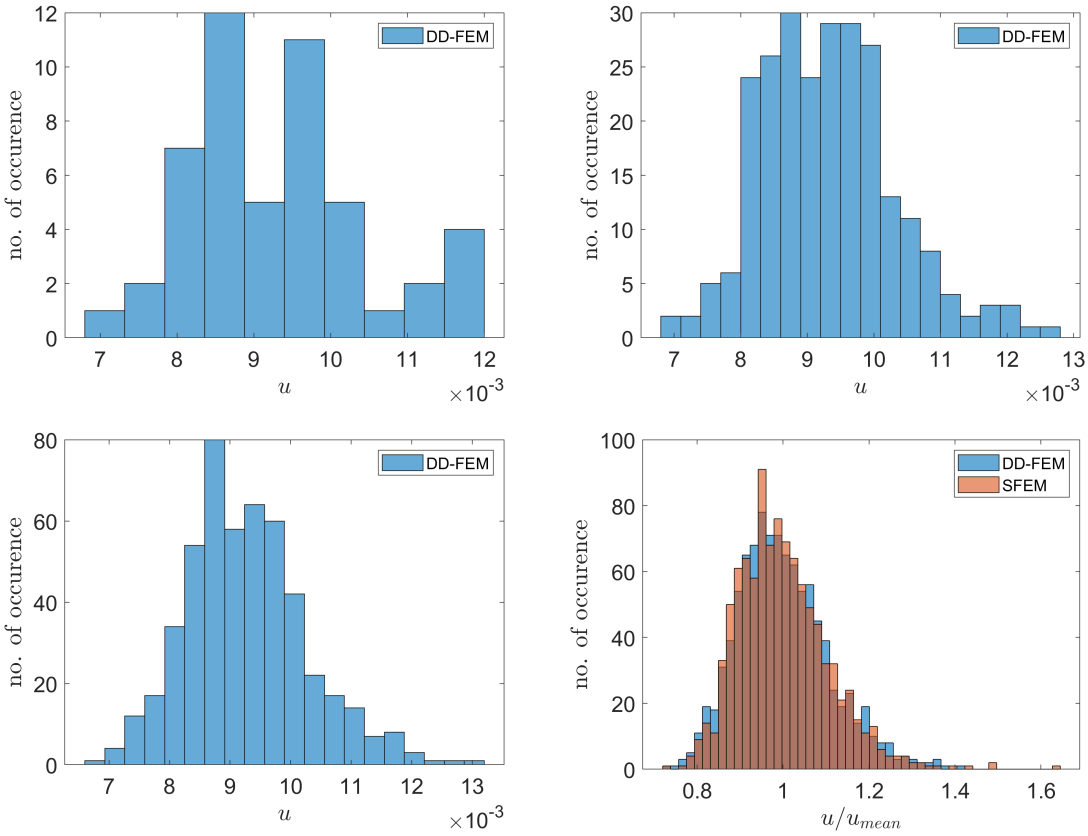


Figure 4.12: Histograms of the maximal displacement for 50, 250 and 500 data sets where one data set is used for all elements. In the last image the distribution is compared to simulations with a stochastic Young's modulus for 1000 data sets.

is one with a stochastic Young's modulus, which is then used as the data set for the whole geometry. Both distributions are centered around their mean as the data-driven solutions had a little shift to the right due to the smaller value of data points. The distributions coincide well, which is no surprise. The data sets as well as the stochastic Young's modulus are generated by the same normal distribution.

Random set for every element

By assigning a random data set to every element the number of different assignments (=number of different simulations) for n data sets and L elements is L^n. This number can get large very fast obviously and that many computations are not even needed to deduce distributions. This also means that n does not have to be such big as before to get the distribution. We only use 20 different data sets here. The results for the

stochastic and the data-driven method are displayed in Fig. 4.13 for 500 computations of each method. First of all we see that the data-driven method again has a little offset in the displacement. Furthermore, we see that the displacement values overall are closer to their mean value. By assigning every element a random value its far more probable that the overall stiffness of the beam is closer to the mean. Centralized, we see that both distributions coincide very well, again showing that the DD-FEM distribution recalls the distribution of the SFEM.

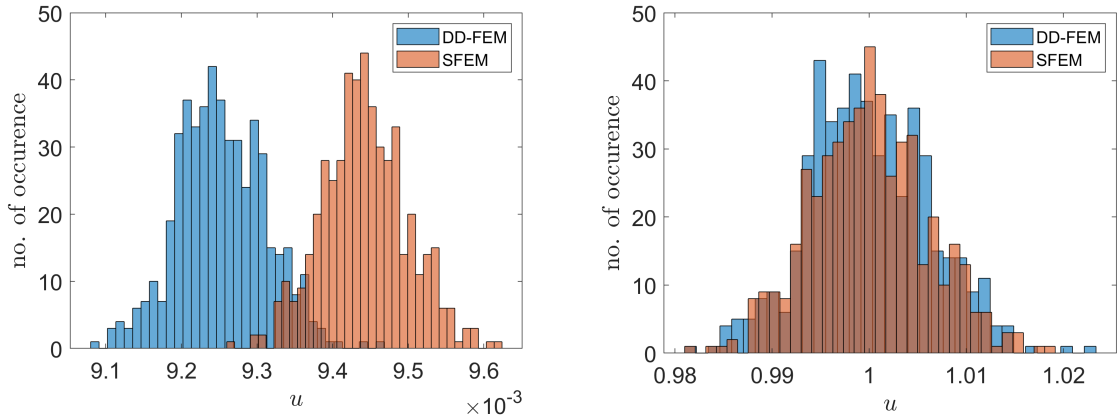


Figure 4.13: Distribution of the maximal displacement if every element gets assigned a different Young's modulus or data set.

Plate with a hole

We also conducted the same procedure to the plate with a hole problem. The distributions of the SFEM and DD-FEM again showed the same behavior as before and did not give us any new insight here and are skipped therefore. We only have a look here onto the stress distribution of the whole part here to see the differences. In Fig. 4.14 the results for the three different finite element methods and the analytic solution are plotted. The data-driven and the stochastic solution directly attract attention with their patchy display of the stress. This comes due to the local different Young's moduli. In one element the stress can be higher due to the decreased stiffness and in the next it may be a little bit lower due to increased stiffness, thus destroying the contours, that can be seen in the finite element and analytic solution. But overall a good agreement of the stress is seen with the addressed patchy parts and small differences at the inner circle where we should remind ourselves that simulations with uncertainties built in are not expected to be equal to an analytic solution.

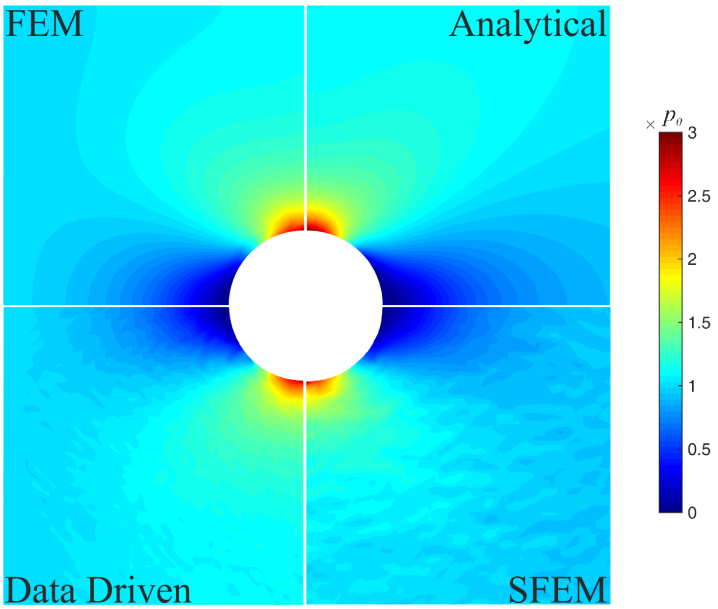


Figure 4.14: Stress σ_x in the plate for the classical FEM, the analytic solution, the DD-FEM and a realization of the SFEM.

4.5 Conclusion of the chapter

In this chapter, we investigated the data-driven finite element method and compared the strategy of considering uncertainties in the material parameters by stochastic fields to the direct use of experimental data in the data-driven analyses. Simple properties of the functionality of the DD-FEM are shown first: Some of them may influence the data-driven method in a negative way. Most of them can be worked around by a good and dense data set. Therefore, they are rather unimportant in the actual literature due to the artificial generation of the data. However, they should not be forgotten for real practical data-driven analysis. The parametric study towards the number of data points and the numerical stiffness showed a good behavior of the DD-FEM. The number of meaningful data points is the main factor influencing the quality of the solution. Solutions showed that 100 data points per strain and stress coefficient lead to satisfying results. The study towards the numerical stiffness showed that the data-driven method is very robust against a wrong estimation of it. Results only showed minor differences and also showed a way to increase precision by analyzing the slope of two or more results.

Last, we investigated the possibility of the DD-FEM to include material uncertainties. The standard way to account for imprecise material data is the SFEM, i.e. the solution of stochastic problems with finite elements whose parameters reflect the randomness.

The mathematical properties of the SFEM regarding accuracy and convergence are well studied and the method is easy to implement in an available finite element software. From the practical point of view, however, the bigger problem is the modeling. Hardly any physical problem follows an obvious stochastic process. Clearly, if the stochastic process is described wrongly, the propagation of the uncertainty through the system cannot be reflected in a correct way. In addition, there are hardly any relevant experimental data for validation available. Most uncertain quantities in mechanical systems are non-Gaussian in nature but the Gaussian model is often used due to its simplicity and the lack of alternative information. This may lead to SFEM computations pretending a predictive capability that is not necessarily covered by the physics. The DD-FEM is a new method that completely bypasses the material modeling. This changes the structure of the finite element equations and results in two coupled sets of linear equations. The implementation requires a new structure but the linearity renders it simple and enables a fast solution. The bottleneck is here the search for the optimal data, which is covered in the next chapter. In particular in three dimensions this makes a data-driven method much more expensive than any standard FEA. In opposite to the SFEM the DD-FEM is not a stochastic method itself. The data-driven solution is deterministic, i.e., it computes for every data set one (not necessarily unique) solution. Histograms were gained by a random assignment of the data sets to the finite elements. From our perspective the data-driven FEA is a new and promising approach to the problem of insufficient material data, which may also be applied to other problems with unclear constitution.

5 Multi-level data set approach

The data-driven finite element simulation is more numerical costly than the classic finite element method. An additional system of equations needs to be solved and many nearest neighbor problems need to be computed. Especially the latter increases the computing time significantly if the number of material points and data points rises. This additional cost needs to be held as low as possible, in particular with a view to realistic three-dimensional problems. In this chapter, we will tackle this problem. The following chapter is based on the publication [75]

5.1 Data search

The core of the data-driven FEA is the search for data pairs $\{\boldsymbol{\epsilon}^{*(j)}, \boldsymbol{\sigma}^{*(j)}\}_i$ in every step of iteration j and at every material point $i = 1, \dots n_m$. At a point i finding the data closest to the computed $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$-values is a nearest neighbor search problem. Formalizing our problem we have: for given a data set $\mathcal{D} \subset \mathbb{R}^k$ and a query point $\mathbf{q} \in \mathbb{R}^k$ find the element $\mathbf{x} \in \mathcal{D}$ such that for all $\mathbf{y} \in \mathcal{D}$ we have: $d(\mathbf{x}, \mathbf{q}) \leq d(\mathbf{y}, \mathbf{q})$. Commonly this is solved by space partitioning algorithms such as grid-file, k-d tree, quadtree or other established solution methods [112]. In the simplest case a binary search algorithm is used, i. e. a tree data structure in which each node has at most two children: the left node is "lesser" than its parent and the right node is "greater" than its parent in a certain dimension. Defining lesser and greater requires to solve a distance d between the data pairs.

A useful data structure for multi-dimensional searches are so-called k-d trees where k is the dimension of the data pairs sorted. Again, each node in a k-d tree has two children, one less and one greater than the parent, but here one dimension at a time is isolated and we iterate through all dimensions. As displayed in Fig. 5.1 the k-d tree splits the data along dimensions into two halves by hyperplanes, i. e. one half of the data has a smaller value in that dimension; the other half has a greater value in that dimension. The new two halves of the data are then split among the next and all remaining dimensions repeatedly. This requires to use the standard greater-than and less-than relations instead of the computation of a distance and is faster. The average running time for such nearest neighbor queries is $\mathcal{O}(\log n)$ [16]. However, for larger dimensions k space partitioning methods reach the worst case scenarios of $\mathcal{O}(n)$. We use here the k-d tree implementation

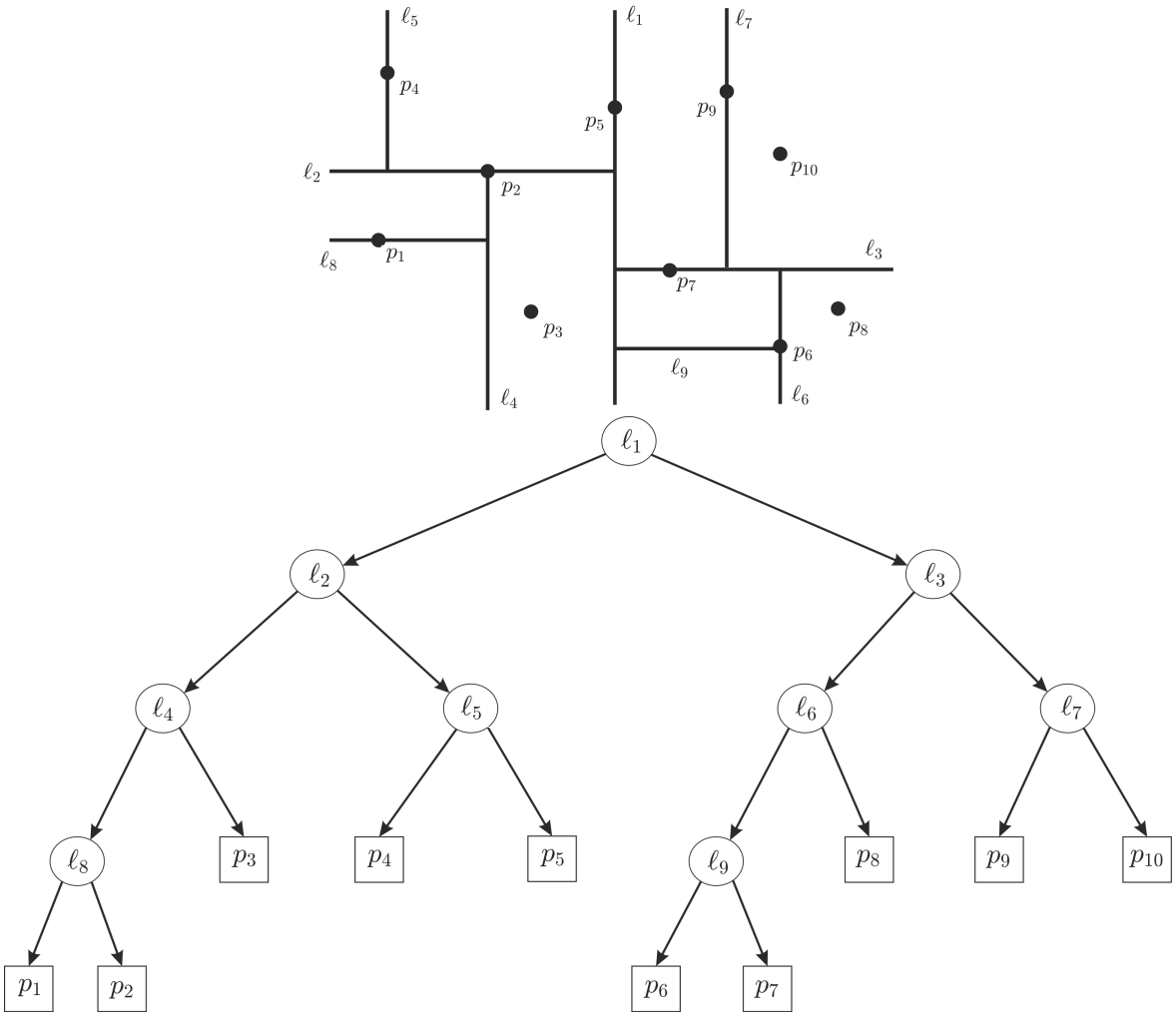


Figure 5.1: Illustration of a k-d tree for a data set with n=10 points; seven hyperplanes ℓ_1 to ℓ_7 separate the points into leaves.

of Matlab, `'knnsearch'`, it searches for the closest K-nearest neighbors by using a k-d tree. Clearly, our query point $\mathbf{q}$ is given by the computed $\{\boldsymbol{\epsilon}, \boldsymbol{\sigma}\}_i$-values whereas $\mathbf{x}$ and $\mathbf{y}$ are the strain-stress pairs of our data set $\mathcal{D}$. In a three-dimensional FEA the dimension of one pair $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$ is $k = 12$.

Numerical costs

From previous data-driven FEAs we know that in the small strain regime about 100 data pairs per dimension provide a sufficient data density of set $\mathcal{D}$ [67, 68, 72]. For $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$ pairs of a three-dimensional FEA this leads to a data set of 100^6 pairs which clearly results in a significant effort for the nearest neighbor queries. Exemplarily we state in Table 5.1

the time needed by `'knnsearch'` to solve the nearest neighbor problem in one iteration for such three-dimensional data sets of different size. The memory size required to store the array is also given. We observe the linear scaling $\mathcal{O}(n)$ and remark that these values are taken from a single search for all material points of the model of Section 5.3; the setup of the tree needs extra time. For example, for the data set with 10^8 pairs the tree setup requires 1008 seconds. These computational costs make it clear that several FEAs with smaller sets will be more favorable than a single simulation with the total data set $\mathcal{D}$.

number of data pairs n	memory size	time in sec.
10^3	96 kB	∼0.012
10^4	960 kB	∼0.032
10^5	9.6 MB	∼0.33
10^6	96 MB	∼3
10^7	960 MB	∼35
10^8	9.6 GB	∼483

Table 5.1: Number of 6-dimensional data pairs per data set, memory size and computational time required by Matlab for the nearest neighbor search based on a model with $N = 19600$ material points.

5.2 Multi-level approach

In order to reduce the computational costs we now expand the data search problem into multiple levels of approximation. On the lowest level a very coarse data set identifies the regions of phase space Z which are needed for problem solution. On further levels those regions are filled with additional data to compute more precise solutions. We remark that other aspects of data reduction like exploiting a presumed material isotropy will not be considered here because such assumptions contradict the idea of a data-driven FEA.

Procedure

The aim of the proposed multi-level approach is to eliminate non-informative or unused data pairs of data set $\mathcal{D}$ and to compute the data-driven FEA on smaller data sets $\mathcal{D}_l \subset \mathcal{D}$, $l = 0, 1, \ldots$. The first step (level 0) is to pick a subset $\mathcal{D}_0 \subset \mathcal{D}$ that consists of a very small number of data pairs but covers the phase space Z as good as set $\mathcal{D}$. With

the reduced data set $\mathcal{D}_0$ the data-driven FEA is computed and the optimal data pairs $(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)_i$ are assigned to all material points $i = 1 \dots n_m$. Consequently, the solution data set on level 0 is given by

$$\mathcal{S}_0 = \{(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)_i \in \mathcal{D}_0 \,,\, i = 1, \dots, n_m\}. \tag{5.1}$$

Obviously $\mathcal{S}_0$ is a subset of $\mathcal{D}_0$ as only data pairs of the input data can be assigned. The approximate solution is coarse and the data pairs of $\mathcal{S}_0$ just indicate the magnitude and direction of the 'real' solution, i. e. the most precisely approximated set $\mathcal{S}$.

In the vicinity of the solution data of set $\mathcal{S}_0$ we now collect data pairs of $\mathcal{D}$ for a new set $\mathcal{D}_1$. This is again done by the norm of the phase space. We define a distance function for two data pairs $(\boldsymbol{\epsilon}_1, \boldsymbol{\sigma}_1)$ and $(\boldsymbol{\epsilon}_2, \boldsymbol{\sigma}_2)$

$$d\left((\boldsymbol{\epsilon}_1, \boldsymbol{\sigma}_1), (\boldsymbol{\epsilon}_2, \boldsymbol{\sigma}_2)\right) = \left(\Psi^e(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2) + \Psi^*(\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2)\right)^{1/2}. \tag{5.2}$$

This energy distance d is now used to define a ball of radius r of data pairs around each value $(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) \in \mathcal{S}_0$. All data pairs within this ball are collected to form the level-1 data set

$$\mathcal{D}_1 = \{(\boldsymbol{\epsilon}, \boldsymbol{\sigma}) \in \mathcal{D} \mid d(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^*, \boldsymbol{\sigma} - \boldsymbol{\sigma}^*) \leq r, \text{ for one } (\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) \in \mathcal{S}_0\}\,. \tag{5.3}$$

The radius r of the ball around $(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) \in \mathcal{S}_0$ is problem specific and depends on the material as well as the data set. In the sense of an efficient strategy it has to hold

$$d_{\min} < r \ll d_{\max} \tag{5.4}$$

where $d_{\min}$ is the minimal distance and $d_{\max}$ the maximal distance of two arbitrary pairs of the data set. In the three-dimensional example of Section 5.3 we use $r = 1.5 d_{\min}$. Alternatively, one may also add just a fixed number of nearest neighbors of each $(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)$.

Now, computing the data-driven algorithm again we gain the solution set $\mathcal{S}_1$ which lead to a better approximation of the real solution

$$\mathcal{S}_1 = \{(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)_i \in \mathcal{D}_1 \,,\, i = 1, \dots, n_m\}. \tag{5.5}$$

For level 2 we may now enrich the set $\mathcal{S}_1$ by $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$ pairs in the vicinity of the assigned data pairs. Repeating this procedure until the wanted accuracy is achieved leads to a solution set $\mathcal{S}_k$ that approaches the data pairs in the total solution set $\mathcal{S}$

$$\mathcal{S}_k = \{(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)_i \in \mathcal{D}_k \,,\, i = 1, \dots, n_m\} \tag{5.6}$$

and the next level data set is

$$\mathcal{D}_{k+1} = \{(\boldsymbol{\epsilon}, \boldsymbol{\sigma}) \in \mathcal{D} \mid d(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^*, \boldsymbol{\sigma} - \boldsymbol{\sigma}^*) \leq r, \text{ for one } (\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) \in \mathcal{S}_k\} \,. \tag{5.7}$$

The number of data pairs used in the initial computation, the number of data pairs which is added to the solution sets and the number of levels depend on the specific problem. With a reasonable choice the number of data pairs in $\mathcal{D}_{k+1}$ is significantly smaller than in the full set $\mathcal{D}$.

Truss structure example

We will exemplarily illustrate the multi-level approach by means of a simple truss structure. The plane truss is displayed in Fig. 5.2a; it consists of 6 rods of stiffness $EA = 6.3 \cdot 10^7\,\mathrm{N}$ and is pulled downwards by a force $F = 1000\,\mathrm{N}$ at the lower right node. For the total data set we use 200 generated strain-stress data pairs describing the material behavior between a stress of -10 MPa and 10 MPa. Needless to say that such a small example can easily be computed with the total data set. The corresponding answer of the system is displayed in Fig. 5.2b magnified by a factor of 100 for visibility.

For level 0 we choose a data set $\mathcal{D}_0 \subset \mathcal{D}$ which is significantly smaller. The data selection can be done randomly, manually or by some sorting algorithm but $\mathcal{D}_0$ has to fully cover the strain-stress space of the problem. Here we chose by random the twenty green data pairs of Fig. 5.2c

$$\mathcal{D}_0 = \{(-3.81 \cdot 10^{-5}, -8.5), \ldots, (3.76 \cdot 10^{-5}, 8.1)\} \,. \tag{5.8}$$

The corresponding data-driven finite element computation is needed to find the regions where data are actually used and assigned. It leads to the black data pairs in Fig. 5.2d which form the set $\mathcal{S}_0$. We remark that also locally different data sets may be used in general.

In the vicinity of the $(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*)$ pairs of $\mathcal{S}_0$ we now collect data pairs for the new set $\mathcal{D}_1$ using definition (5.3). Here $\mathcal{D}_1$ comprises all data pairs inside a ball of radius $r = 1\,\mathrm{Nm}$ around the solution pairs.

The next data-driven computation leads to possible new data pairs which form the solution data set $\mathcal{S}_1$ in the neighborhood of the prior ones. One neighborhood with the new data point is shown in Fig. 5.2f. Typically they are more precise than the solution of set S_0. This procedure can be repeated as often as wanted. Usually the data sets enlarge during computation such that $|\mathcal{D}_1| \leq |\mathcal{D}_2| \leq \ldots |\mathcal{D}_k| \leq |\mathcal{D}|$ but are still smaller than the original data set $\mathcal{D}$. The remaining task is to decide whether the solution has converged i.e. the final solution set $\mathcal{S}$ has been found. Because an analytical solution is usually unknown a relative error may serve as a termination criterion. Typically the global value of error functional (2.30) is used, eq. (2.31). Another way to decide

the multi-level approximation to terminate is the case that in subsequent iterations the same data pairs are assigned such that $\mathcal{S}_k = \mathcal{S}_{k-1}$.

5.3 Computation of a sandwich panel connection

Clearly, the benefit of the proposed method increases the bigger the data sets and the finite element model are. The latter is determined by geometry, approximation order and mesh size whereas the number of data pairs especially increases with the dimension. Therefore we want to apply the proposed multi-level approach to a fully three-dimensional engineering problem.

Geometry and loading

Our problem of interest is the stress state in a T-joint connection of two composite honeycomb sandwich panels, see Fig. 5.3. In a sandwich panel the combination of a lightweight core with adjacent face sheets of different material provides a coupled mechanical response which is difficult to predict, cf. [64, 65]. In our case the sandwich is made of a hexagonal honeycomb core with a thickness of 19.05 mm and covering face sheets of pre impregnated fibers (prepregs) of 0.28 mm thickness. Basis material of the core is resin impregnated Nomex aramid paper, the prepregs consist of two different layers of glass fibers in phenolic resin. For the connection of web and base plate of the T-joint specimen a polyurethane adhesive is used; more details can be found in [65].

Clearly, a classical finite element computation of such a structure requires either a very finely resolved finite element mesh or homogenized data for the specific sandwich composition. We remark that the homogenized material cannot be described by a simple Hookean law since it is orthotropic with direction dependent differences in its compression-tension behaviour and hardly any shear resistance.

In the following we analyze two loading scenarios, namely a bending and a tensional force on the web. Tension, i.e. pulling the web from the base, gives a rather simple loading regime of the T-joint but corresponds to our experiments in [65]. Bending of the web is added here to compare the stress state with a reference finite element analysis.

Bending of the T-joint

We start with the latter, namely a distributed force of $8.8 \cdot 10^{-3}\,\mathrm{N/mm^2}$ in horizontal direction as indicated in Fig. 5.4. The T-joint is clamped at the bottom side. We refer to a data set $\mathcal{D}$ of effective strain-stress pairs for the sandwich panel based on experimental data and generated with a homogenized elastic modulus of 265 MPa. The

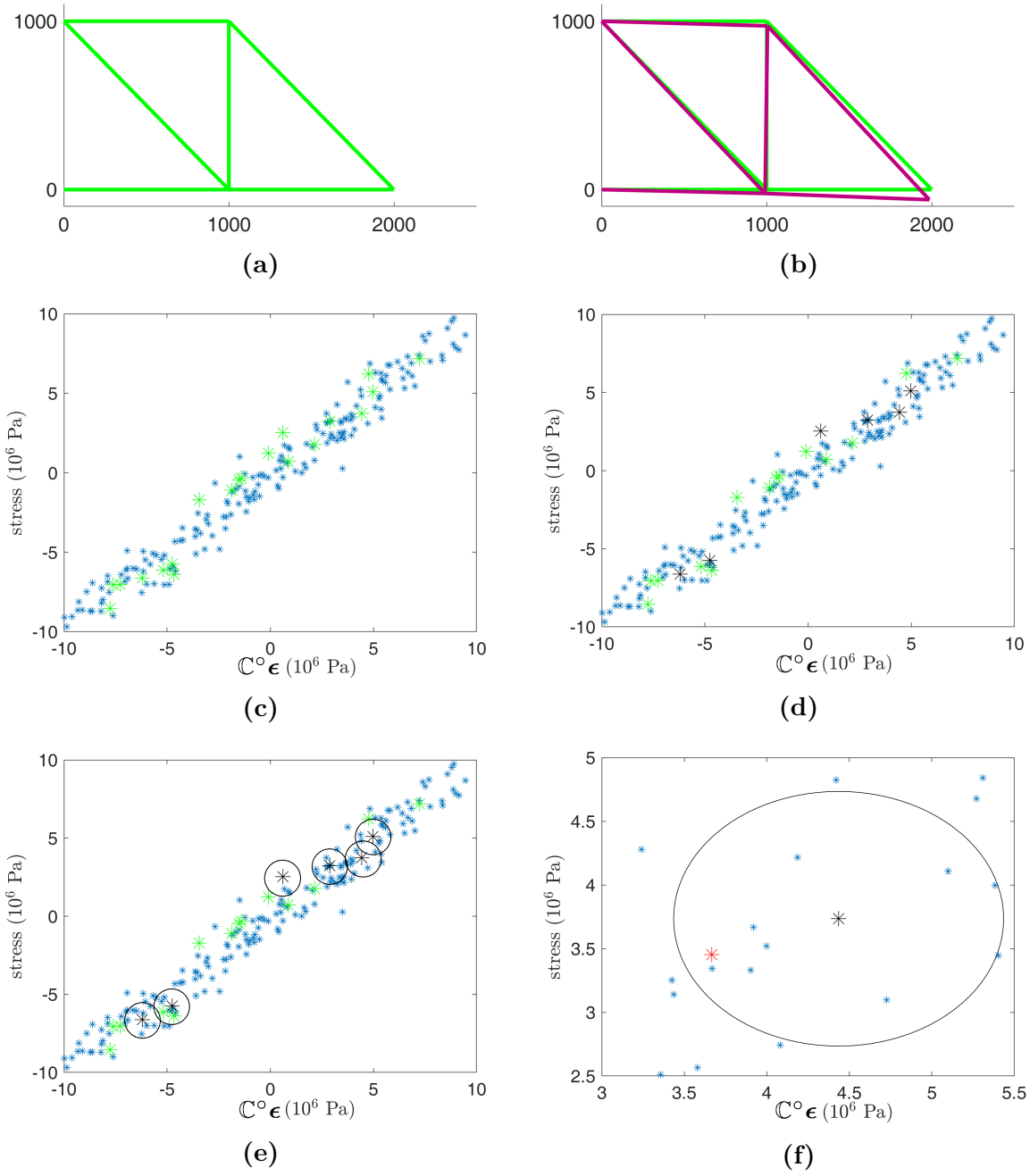


Figure 5.2: Workflow of the multi-level method (the strains are scaled with $\mathbb{C}^\circ = 2.1 \cdot 10^5$ MPa for readability): (a) Undeformed truss structure, (b) Deformed truss structure, (c) Full data set $\mathcal{D}$ (blue) and data set $\mathcal{D}_0$ of level 0 (green), (d) Solution data points $\mathcal{S}_0$ assigned in level 0 (black), (e) Neighbourhood of the solution data pairs $\mathcal{S}_0$ forming the next level data set $\mathcal{D}_1$, (f) Zoom into the neighbourhood of one assigned data pairs of $\mathcal{S}_0$, data of $\mathcal{D}_1$ and solution $\mathcal{S}_1$ (red).

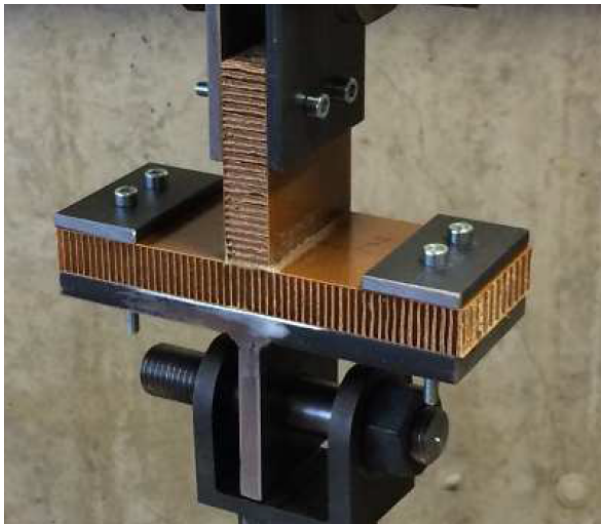


Figure 5.3: The sandwich panel T-joint during the tension test of [65].

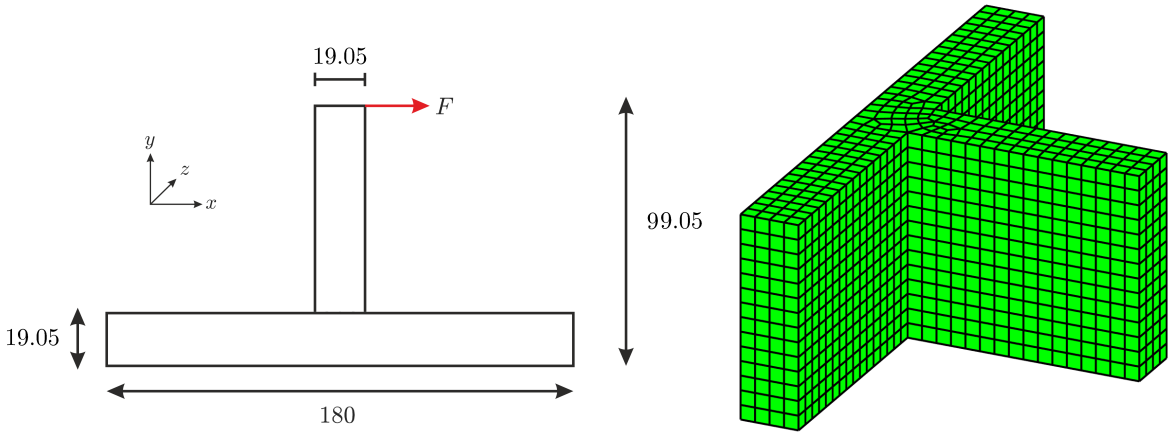
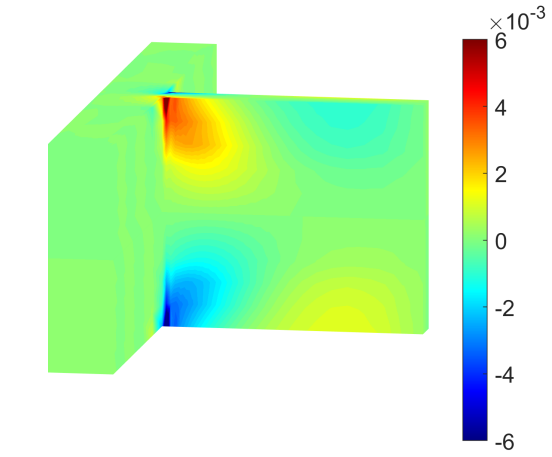


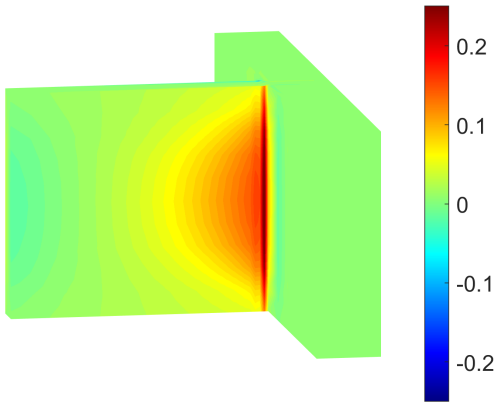
Figure 5.4: Geometry of the T-joint and finite element mesh: The width of the specimen is 70 mm. The mesh consists of 2450 hexahedron elements, each with 8 gaussian points. The applied force per upper surface is $8.8 \cdot 10^{-3}\,\mathrm{N/mm^2}$.

glued connection is assumed to be stiff. Preliminary simulations showed that in the small strain regime a number of 100 data pairs per strain and stress component are sufficient.

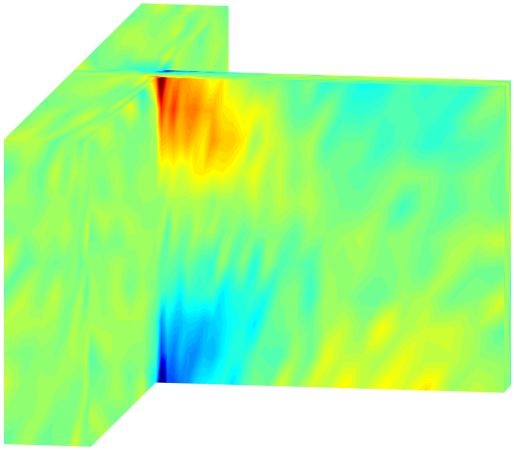
Consequently, we assume a data set $\mathcal{D}$ of $100^6 = 10^{12}$ data pairs to fully cover the strain-stress space of the problem, i.e. this data-pair density is able to give sufficiently accurate results. For the simulation the data set was sampled from a given grid of the stress space. Corresponding strain values were computed with a hidden material law. For our multi-level computation we start with an initial subset $\mathcal{D}_0$ of 25^6 data pairs. These pairs are chosen randomly out of the full set $\mathcal{D}$ and comprise 0.0244% of the



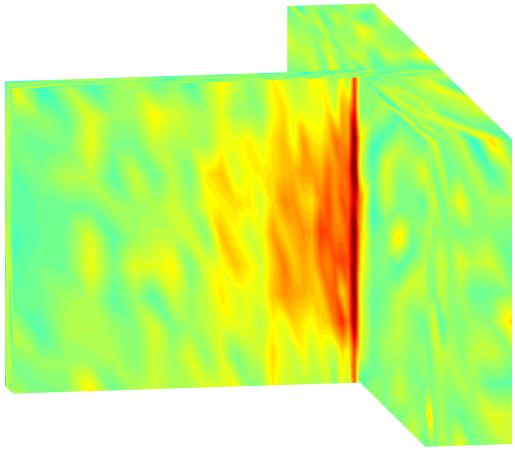
(a) Reference FEA: τ_{yz}



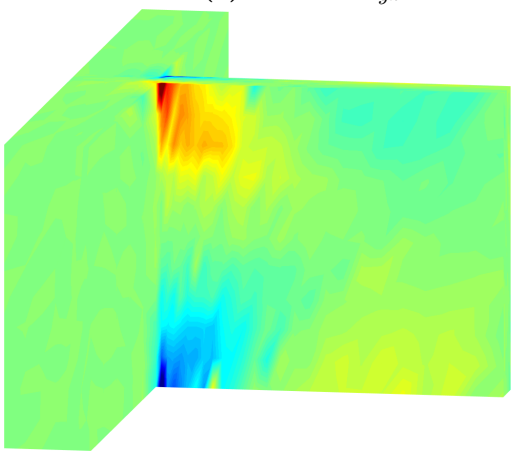
(b) Reference FEA: σ_z



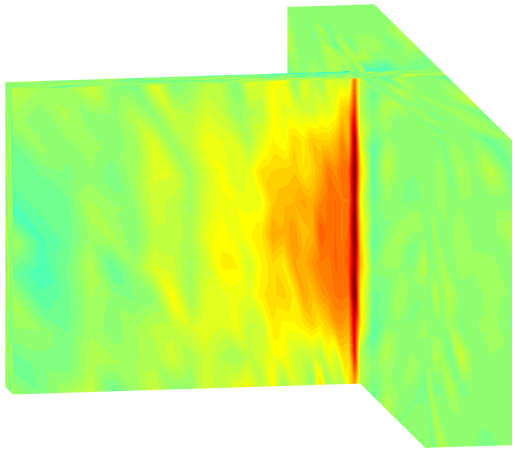
(c) Level 0: τ_{yz}



(d) Level 0: σ_z

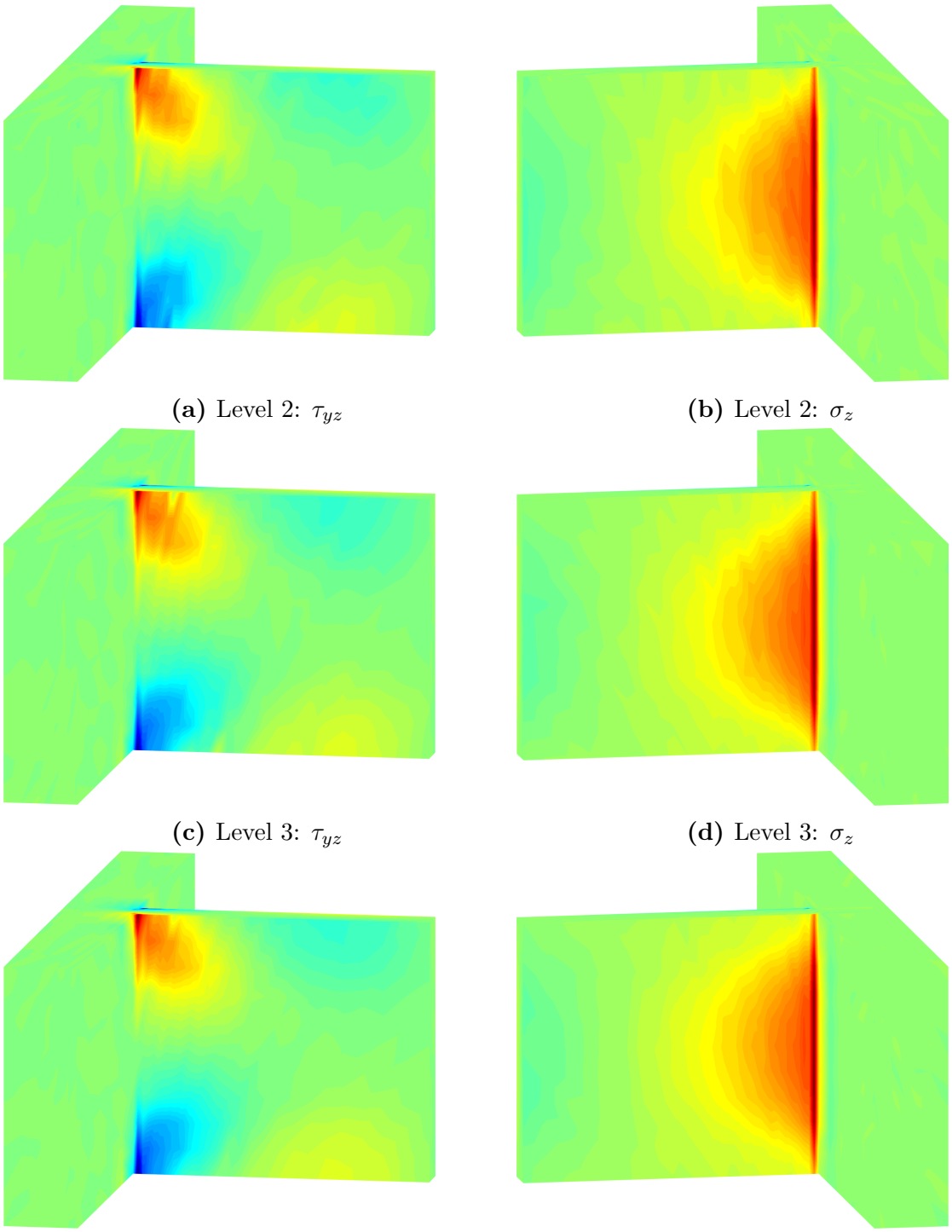


(e) Level 1: τ_{yz}



(f) Level 1: σ_z

Figure 5.5: Bending of the T-joint: shear stress τ_{yz} and normal stress σ_z of a reference FEA with $E = 265\,\text{MPa}$ and $\nu = 0.2$ (first row) and corresponding solutions of level 0 and level 1. The color scheme shows the stress in MPa and applies to all images.



(a) Level 2: τ_{yz} **(b)** Level 2: σ_z

(c) Level 3: τ_{yz} **(d)** Level 3: σ_z

(e) Level 4: τ_{yz} **(f)** Level 4: σ_z

Figure 5.6: Bending of the T-joint: shear stress τ_{yz} and normal stress σ_z of level 2 to 4 of the multi-level method; color scheme from Fig. 5.5.

Level l	data pairs in $\mathcal{D}_l$	data pairs in $\mathcal{S}_l$	RMS
0	244.140.625	10805	0,3366
1	7.876.845	14.415	0,0747
2	10.508.535	15.535	0,0571
3	11.325.015	16.034	0,0541
4	11.688.786	16.376	0,0523

Table 5.2: Results of the multi-level method for the T-joint on different levels.

original data. Fig. 5.5 illustrates the results. In the upper line two stress components of the reference FEA are displayed, namely the shear stress τ_{yz} and the tension of the web σ_z. In the second line the corresponding results computed with the data set $\mathcal{D}_0$ are shown (level 0). Clearly, this simulation gives only a rough representation of the stress distribution because the distances of the few data are quite large and so the stress distributions in Fig. 5.5c and 5.5d are rather patchy. To quantify the differences between the two solution we evaluate the root mean square error for the strains

$$\boldsymbol{\epsilon}_{RMS} = \left(\frac{\int_\Omega \Psi^e(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\text{ref}})\, \mathrm{d}\Omega}{\int_\Omega \Psi^e(\boldsymbol{\epsilon}_{\text{ref}})\, \mathrm{d}\Omega} \right)^{1/2} . \tag{5.9}$$

In Table 5.2 the error (4.1), the number of data pairs in the solution and the input data set are displayed for the different levels. For the simulation of level 0 we see a relative high error of over 30%. At the 19600 material points of the finite element model, a computation with the data of $\mathcal{D}_0$ results in a solution data set $\mathcal{S}_0$ with only 10805 data pairs. For the next levels the data sets are expanded by the data pairs closer than $r = 1.5 d_{\min}$, eq. (5.4). This expansion is easy when the data are given in a grid-like arrangement. Otherwise a proximity search with the whole data set needs to be conducted once. In this way, the computation on level 1 with $\mathcal{D}_1$ uses 7 876 845 data pairs, i.e. $7.88 \cdot 10^{-4}\%$ of the original data. The corresponding computations are fast and the needed amount of memory is still small.

The stress distributions for all four computed levels are shown in Fig. 5.5 and 5.6. The contours of the stress get smoother with every level and the last level visually coincides with the reference finite element solution. The number of data pairs in the solution data sets increases with every iteration $|\mathcal{D}_1| \leq \cdots \leq |\mathcal{D}_4|$. This allows a more suitable assignment of the data pairs and so the error reduces down to $\sim 5\%$ which is a satisfying result for a fully three-dimensional computation. We remark that in the plane stress computations of [67, 72] the errors are in the same order of magnitude.

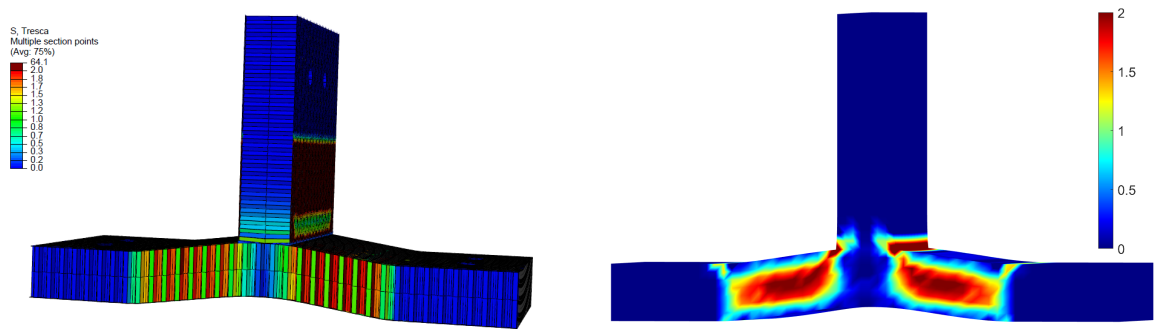


Figure 5.7: Distribution of the maximal shear stress in an Abaqus computation of Fig. 3 [65] (left) and data-driven computation with 10 816 223 data pairs on level 5 of the multi-level method (right).

Tension on the web

In the second load case the web is pulled and the base is partially fixed at the left and right quarter, see Fig. 5.3. The results of a data-driven computation with 5 levels of data sets are displayed in Fig. 5.7 and compared to a FEA of [65] resolving all sandwich details. The deformation of the specimen is magnified here by a factor of 2.5. Despite of a simplified modeling regarding the glued connection, we see a good agreement of the distributed shear stress.

In Fig. 5.8 the load is displayed versus the displacement at the top of the T-joint for both, the experiment and the data-driven FEA. We remark that in the experiment the specimen was loaded till rupture, i.e. decohesion of the glued web-base connection. Such failure is not computed here but as long as the T-joint is intact the numerical results of the load-displacement curve coincide well with the experimental results. The results diverge for the beginning of damage in the later loading stage because this was not covered by our total data set $\mathcal{D}$. Summarizing we remark that our final set $\mathcal{D}_4$ has 10 816 223 data pairs compared to the 10^{12} data pairs of set $\mathcal{D}$ and gives sufficiently accurate results.

5.4 Conclusion of the chapter

In this chapter, a multi-level method to reduce the computational costs for data-driven finite element analyzes with big data sets is proposed. With the introduced strategy, the number of data pairs considered in a computation can significantly be reduced, which will make it easier to conduct complex three-dimensional data-driven simulations.

Starting with a very coarse initial data set $\mathcal{D}_0$ as a subset of the total data set $\mathcal{D}$ the corresponding data-driven FEA gives a rough first approximation of the solution. Then,

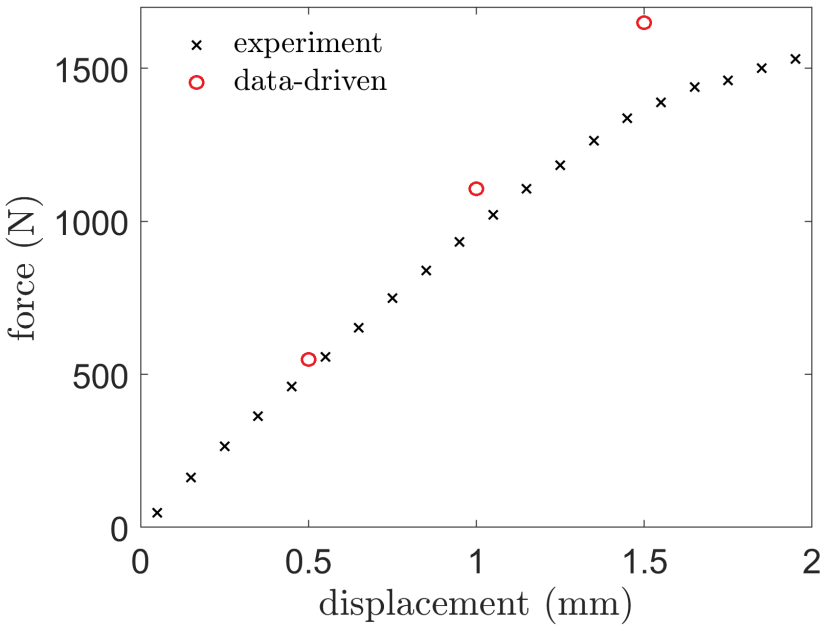


Figure 5.8: Load-displacement curve of the data-driven FEA and the experimentally obtained values of [65].

the next level data sets are successively refined and adapted to the solution. With repeated simulations, we obtain an accurate solution using just a subset of the original data. From an illustrating engineering example we see how computational time and demanded storage are reduced by the multi-level method. Instead of a full three dimensional simulation with 10^{12} data pairs the simulation is decomposed into five smaller simulations with only around 10^7 data pairs. Although the stress state of the T-joint is rather simple, it illustrates the potential of the method. The decreasing approximation error nicely shows the increase in the quality of the solutions at increasing levels.

We see the future of the data-driven FEA in the simulation of complex materials which do not follow a simple constitutive relation. Here it will be hard to reduce the dimension of the data phase space without loss of information. Furthermore, history or time-dependent material behaviors will require the data base to cover a wide range of situations. In such a context, it will be necessary to reduce the computational costs of data assignment by the proposed adaptive multi-level method.

6 Data-driven computations with RVE generated data sets

The material data sets were generated synthetically in most previous publications on DD-FEM [34, 67, 88]. In this chapter, we suggest utilizing stochastic representative volume elements (RVEs) to generate a material database required for data-driven computations either a priori or on the fly. In order to provide the data set $\mathcal{D}$, the stochastic RVE representing a specific foam must be subjected to all sorts of deformation states. From the solution of these microscopic BVPs, the corresponding stress states are obtained. In such a way, we generate the state space tuples $\boldsymbol{z}_i$ that describes the macroscopic behavior of the heterogeneous material. This chapter partially relies on the publication [74].

Computational material testing

For solid mechanic problems, the material data set which replaces the constitutive model needs to represent strains and stresses in the dimension of the problem. They have to encompass an almost infinite number of data tuples, covering every possible mechanical state under consideration. Such data sets can be gained experimentally, using in-situ computed tomography for example, but these sorts of measurements are expensive and for many problems not available. An alternative way of data acquisition can be the use of computational 'material testing' in the sense of sample calculations for the heterogeneous material. In the classical approach, representative volume elements, which allow for a detailed description of the microscopic characteristics, are used to derive effective material properties [8]. For the finite element analysis (FEA) of engineering components this technique leads directly to computational homogenization and FE^2 computations [86, 39], where the two- or multi-scale problem is decoupled into nested subproblems, i.e. a microscopic BVP and a macroscopic BVP. For the macroscopic BVP to be solved, the required effective quantities are obtained from the solutions of the many microscopic BVPs at the material points of the discretized continuum, [7, 23].

Here, we focus specifically on an open-cell elastomeric foam. Solid foams, typically used for thermal insulation, vibration damping and sandwich-structured composites,

are lightweight engineering materials with a heterogeneous cellular microstructure. Their design and computation require either strong simplifications or sophisticated material models. To avoid these, a direct use of measurement data in the computation would be desirable. For foam-like materials, homogenization strategies are chosen in [58, 84] for example. The mechanical properties of such foams are determined by the cellular microstructure and by the material's properties. In our study, the foam is made of polyurethane (PUR), a common elastomer produced by a reaction of an isocyanate with a polyol in the presence of a catalyst and some additives. Adding a certain amount of water to the reaction produces CO_2, which results in the formation of gaseous bubbles, [57, 15]. After solidification, a foam with defined porosity but an irregular cellular microstructure is 'baked'. This variability can be captured by stochastically generated RVEs, based on the same statistical parameters but with different expressions of the characteristics from one realization to another.

6.1 Non-linear DD-FEM

We consider a solid of domain Ω_0 in its reference configuration and deforming to the current configuration under the action of external body forces $\rho_0 \boldsymbol{B}$ and boundary tractions $\boldsymbol{T}$; the fields in capitals refer to the reference configuration. The solid's deformation $\boldsymbol{\varphi}(\boldsymbol{X}) : \Omega_0 \mapsto \mathbb{R}^3$ is completely described by the deformation gradient

$$\boldsymbol{F} = \nabla \boldsymbol{\varphi}(\boldsymbol{X}) \,. \tag{6.1}$$

The work conjugate stress is the first Piola-Kirchhoff tensor $\boldsymbol{P}$; the second Piola-Kirchhoff tensor follows via $\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P}$. The stresses fulfill the linear and the angular momentum balances

$$\operatorname{Div} \boldsymbol{P} + \rho_0 \boldsymbol{B} = 0 \qquad \text{in } \Omega_0 \tag{6.2}$$

$$\boldsymbol{P}^T \boldsymbol{F} - \boldsymbol{F} \boldsymbol{P}^T = 0 \tag{6.3}$$

and consequently is $\boldsymbol{S} = \boldsymbol{S}^T$. The solid is subjected to geometrical and static boundary conditions at its boundaries Γ_0 and Γ_1 with outward unit normal $\boldsymbol{N}$,

$$\boldsymbol{\varphi} = \bar{\boldsymbol{\varphi}} \qquad \text{on } \Gamma_0 \,, \tag{6.4}$$

$$\boldsymbol{P}\,\boldsymbol{N} = \boldsymbol{T} \qquad \text{on } \Gamma_1 \,. \tag{6.5}$$

where $\Gamma_0 \cup \Gamma_1 = \partial\Omega_0$ and $\Gamma_0 \cap \Gamma_1 = \emptyset$. These physical equations determine the set $\mathcal{C}$ of mechanical admissible strain-stress states. The goal of the data-driven problem is to find the minimal distance between this constraint set and the material data set, as stated by eq. (**??**). The material data set $\mathcal{D}$ consists always of tuples of a strain measure and a stress measure. In this paper we will use three different pairings: the linear-elastic

strain and stress $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$, the deformation gradient together with the first Piola-Kirchhoff stress $(\boldsymbol{F}, \boldsymbol{P})$, and the right Cauchy-Green tensor with the second Piola-Kirchhoff stress $(\boldsymbol{C}, \boldsymbol{S})$.

6.1.1 Data-driven problem in F, P

To formulate the boundary value problem (6.2 - 6.5) in the data-driven form, a proper finite deformation metric is needed. Here we follow [**?**] where, for a convex function Ψ^e and its convex conjugate Ψ', a distance of the form

$$d(\boldsymbol{z}, \boldsymbol{z}') = \int_\Omega \Psi^e(\boldsymbol{F} - \boldsymbol{F}') + \Psi'(\boldsymbol{P} - \boldsymbol{P}')\,\mathrm{d}\Omega \tag{6.6}$$

is proposed. For the specific choice of

$$\Psi^e(\boldsymbol{F}) = \frac{\mu}{2}(\boldsymbol{F} : \boldsymbol{F}) \qquad \text{and} \qquad \Psi'(\boldsymbol{P}) = \frac{1}{2\mu}\boldsymbol{P} : \boldsymbol{P}\,, \tag{6.7}$$

the data-driven problem (**??**) requires to minimize the distance (6.6) and to fulfil the constraints of static equilibrium (6.2) and angular momentum balance (6.3). This gives the problem

$$\begin{aligned} W = \min_{(\boldsymbol{F}', \boldsymbol{P}') \in \mathcal{D}} \int_\Omega & \left(\Psi^e(\boldsymbol{F} - \boldsymbol{F}') + \Psi'(\boldsymbol{P} - \boldsymbol{P}')\right) \\ & + \boldsymbol{\lambda}\left(\nabla \cdot \boldsymbol{P} + \rho_0 \boldsymbol{B}\right) + \boldsymbol{\lambda}_2 : \left(\boldsymbol{F}\boldsymbol{P}^T - \boldsymbol{P}^T\boldsymbol{F}\right) \mathrm{d}\Omega \;\rightarrow \text{optimum} \end{aligned} \tag{6.8}$$

where $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}_2$ are Lagrange multiplier fields. The corresponding data set $\mathcal{D} = \{(\boldsymbol{F}', \boldsymbol{P}')_1, \ldots, (\boldsymbol{F}', \boldsymbol{P}')_n\}$ contains tuples of the deformation gradient and the first Piola-Kirchhoff stress tensor. We presume them to be physical meaningful and, specifically, to fulfil eq. (6.3). The parameter μ calibrates units and magnitudes and is purely numerical in nature; we therefore write μ_0 in the following.

Functional (6.8) states a non-linear data-driven boundary value problem that could be solved numerically. The additional Lagrange multiplier field $\boldsymbol{\lambda}$ enforcing (6.2) is a displacement-like vector field; field $\boldsymbol{\lambda}_2$ enforcing (6.3) is tensor-valued. Because the unknown fields are coupled and the formulation cannot be solved explicitly, problem (6.8) cannot be treated with a standard finite element approach. One way to simplify it is to *not* enforce the angular momentum, presuming a negligible distance between the data of solution $\mathcal{C}$ and the material data $\mathcal{D}$ in that respect. However, in that way the problem is linearized and, after discretization, we obtain the same set of finite element equations like in linear elasticity. For a derivation we refer to our previous work [72]; the result is stated in Appendix 1.

6.1.2 Data-driven problem in C, S

For a non-linear finite strain formulation we proceed formulating the problem with deformation measures fulfilling the angular momentum balance a priori. We use the left Cauchy-Green tensor $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ and the corresponding second Piola-Kirchhoff stress tensor $\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P}$ and, similar to (6.7), the functions

$$\Psi^e(\boldsymbol{C}) = \frac{\mu}{2}\left(\boldsymbol{C} : \boldsymbol{C}\right) \quad \text{and} \quad \Psi'(\boldsymbol{S}) = \frac{1}{2\mu}\boldsymbol{S} : \boldsymbol{S}\,, \tag{6.9}$$

to define the distance. This gives the functional to be optimized

$$W = \min_{(\boldsymbol{C}',\boldsymbol{S}')\in\mathcal{D}} \int_\Omega \left(\Psi^e(\boldsymbol{C}-\boldsymbol{C}') + \Psi'(\boldsymbol{S}-\boldsymbol{S}')\right) + \boldsymbol{\lambda}\left(\nabla\cdot\boldsymbol{F}\boldsymbol{S} + \rho_0\boldsymbol{B}\right)\,\mathrm{d}\Omega \tag{6.10}$$

where the data set now contains $(\boldsymbol{C}, \boldsymbol{S})$ tuples, $\mathcal{D} = \{(\boldsymbol{C}', \boldsymbol{S}')_1, \ldots, (\boldsymbol{C}', \boldsymbol{S}')_n\}$. The distance minimization has to be performed under the equilibrium constraint (6.2), i.e. $\nabla\cdot(\boldsymbol{F}\boldsymbol{S}) + \rho_0\boldsymbol{B} = 0$. Expressing $\boldsymbol{C}$ with the displacement field $\boldsymbol{u}$

$$\boldsymbol{C} = \nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T + \nabla\boldsymbol{u}^T\nabla\boldsymbol{u} + \boldsymbol{I} \tag{6.11}$$

we obtain the functional to be optimized in terms of $\boldsymbol{u}$, $\boldsymbol{S}, \boldsymbol{\lambda}$. For optimal data tuples $(\boldsymbol{C}^*, \boldsymbol{S}^*)$ it reads

$$\begin{aligned} W(\boldsymbol{u},\boldsymbol{S},\boldsymbol{\lambda}) = \int_\Omega \Big(&\frac{\mu_0}{2}\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T + \nabla\boldsymbol{u}^T\nabla\boldsymbol{u} + \boldsymbol{I} - \boldsymbol{C}^*\right)^2 : \boldsymbol{I} \\ &+ \frac{1}{2\mu_0}(\boldsymbol{S}-\boldsymbol{S}^*)^2 : \boldsymbol{I} + \boldsymbol{\lambda}\cdot(\nabla\cdot(\boldsymbol{I}+\nabla\boldsymbol{u})\boldsymbol{S} + \rho_0\boldsymbol{B})\Big)\,\mathrm{d}\Omega\,. \end{aligned} \tag{6.12}$$

A variation with respect to the displacement field, $\delta_{\boldsymbol{u}}W = 0$, gives after some rearrangement,

$$\begin{aligned} 2\mu_0\int_\Omega \left(\nabla\delta\boldsymbol{u} + \nabla\delta\boldsymbol{u}^T\nabla\boldsymbol{u}\right) : \left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T + \nabla\boldsymbol{u}^T\nabla\boldsymbol{u} + \boldsymbol{I} - \boldsymbol{C}^*\right)\mathrm{d}\Omega \\ = \int_\Omega \nabla\boldsymbol{\lambda} : \nabla\delta\boldsymbol{u}\,\boldsymbol{S}\,\mathrm{d}\Omega\,, \end{aligned} \tag{6.13}$$

and $\delta_{\boldsymbol{\lambda}}W = 0$ results in

$$\int_\Omega \nabla\delta\boldsymbol{\lambda}\left(\boldsymbol{I} + \nabla\boldsymbol{u}\right)^T : \boldsymbol{S}\,\mathrm{d}\Omega = \int_\Omega \delta\boldsymbol{\lambda}\cdot\boldsymbol{T}\,\mathrm{d}\Omega + \int_\Omega \delta\boldsymbol{\lambda}\cdot\rho_0\boldsymbol{B}\,\mathrm{d}\Omega =: \boldsymbol{f}^{\mathrm{ext}}\,. \tag{6.14}$$

Finally, from $\delta_{\boldsymbol{S}}W = 0$, we derive the expression for the stresses,

$$\boldsymbol{S} = \boldsymbol{S}^* + \mu_0\nabla\boldsymbol{\lambda}(\boldsymbol{I} + \nabla\boldsymbol{u})^T \tag{6.15}$$

which now depend on the displacements and the multiplier field and are clearly non-linear. Inserting eq. (6.15) into eq. (6.14), it results

$$\int_\Omega \nabla\delta\boldsymbol{\lambda}(\boldsymbol{I}+\nabla\boldsymbol{u})^T : \left(\boldsymbol{S}^* + \mu_0\nabla\boldsymbol{\lambda}(\boldsymbol{I}+\nabla\boldsymbol{u})^T\right) \mathrm{d}\Omega = \boldsymbol{f}^{\mathrm{ext}}\,. \tag{6.16}$$

The eqs. (6.13) and (6.16) form the wanted coupled and non-linear system of equations for the fields $\boldsymbol{u}$ and $\boldsymbol{\lambda}$. We summarize them as residual equations,

$$\begin{aligned}\int_\Omega \nabla\delta\boldsymbol{u}^T : \Big(&2\mu_0\boldsymbol{F}(\boldsymbol{u})^T\Big(\boldsymbol{F}(\boldsymbol{u})^T\boldsymbol{F}(\boldsymbol{u}) - \boldsymbol{C}^*\Big) \\ &-\nabla\boldsymbol{\lambda}^T\boldsymbol{S}^* - \mu_0\nabla\boldsymbol{\lambda}^T\nabla\boldsymbol{\lambda}\boldsymbol{F}(\boldsymbol{u})^T\Big)\,\mathrm{d}\Omega = 0\end{aligned} \tag{6.17}$$

$$\int_\Omega \nabla\delta\boldsymbol{\lambda} : \left(\mu_0\boldsymbol{F}(\boldsymbol{u})\nabla\boldsymbol{\lambda}\boldsymbol{F}(\boldsymbol{u})^T + \boldsymbol{F}(\boldsymbol{u})\boldsymbol{S}^*\right)\mathrm{d}\Omega - \boldsymbol{f}^{\mathrm{ext}} = 0\,. \tag{6.18}$$

where we have written $\boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{I} + \nabla\boldsymbol{u}$ for the sake of clarity.

6.1.3 Finite element formulation in C, S

For the numerical simulation we decompose the domain Ω_0 of dimension d into finite elements Ω_e,

$$\Omega_0 \approx \tilde{\Omega}_0 = \bigcup_{\mathcal{E}} \Omega_e$$

where $\mathcal{E} = \{1, \ldots, n_{el}\}$ is the set of all elements and $\cup$ abbreviates the assembly. We employ the usual ansatz for the unknown fields and their variations

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{x}) &\approx \tilde{\boldsymbol{u}}(\boldsymbol{x}) = \mathbf{N}\hat{\boldsymbol{u}} & \delta\tilde{\boldsymbol{u}}(\boldsymbol{x}) &= \mathbf{N}\,\delta\hat{\boldsymbol{u}} \\ \boldsymbol{\lambda}(\boldsymbol{x}) &\approx \tilde{\boldsymbol{\lambda}}(\boldsymbol{x}) = \mathbf{N}\hat{\boldsymbol{\lambda}} & \delta\tilde{\boldsymbol{u}}(\boldsymbol{x}) &= \mathbf{N}\,\delta\hat{\boldsymbol{\lambda}}\end{aligned} \tag{6.19}$$

where the matrix $\mathbf{N}$ contains the shape functions N_k for every degree of freedom $k = 1, \ldots, n_{\mathrm{dof}}$. The vectors $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\lambda}}$ denote the nodal displacements and Langrange parameters. The gradients are summarized in matrix $\mathbf{B} \equiv \nabla\mathbf{N}$; the subscript e refers to one element. The finite element discretization of the system (6.17-6.18) is performed with the ansatz (6.19) and the deformation gradient is calculated in each integration point as

$$\mathbf{F}^e = \mathbf{I} + \mathbf{B}^e\hat{\boldsymbol{u}}^e$$

where $\mathbf{F}^e$ has the suitable vector form. The discretized system of residual equations is then:

$$\begin{aligned}\mathbf{R}_u &= \int_{\Omega_e} \mathbf{B}^{eT}\left(2\mu_0\mathbf{F}^{\mathbf{e}T}\left(\mathbf{F}^{\mathbf{e}T}\mathbf{F}^{\mathbf{e}} - \mathbf{C}^{*}\right) - \mathbf{B}^{eT}\hat{\boldsymbol{\lambda}}\mathbf{S}^{*} - \mu_0\mathbf{B}^{eT}\hat{\boldsymbol{\lambda}}\mathbf{B}^{e}\hat{\boldsymbol{\lambda}}\mathbf{F}^{eT}\right)\mathrm{d}\Omega = 0 \\ \mathbf{R}_\lambda &= \int_{\Omega_e} \mu_0\mathbf{B}^{eT}\mathbf{F}^{e}\mathbf{B}^{e}\hat{\boldsymbol{\lambda}}\mathbf{F}^{eT} + \mathbf{B}^{eT}\mathbf{F}^{e}\mathbf{S}^{*}\,\mathrm{d}\Omega - \mathbf{f}^{e} = 0\end{aligned} \tag{6.20}$$

The solution of the system (6.20) requires an iterative scheme, typically a Newton-Rhapson iteration. The resulting finite element expressions are given in Appendix 2.

We remark that a derivation of the DD algorithm for strain $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$ and stress $\boldsymbol{S}$, which results in a similar system of equations, is presented in [**?**].

6.2 Recording of the material data sets

The material data sets required for the DD-FEM are gained from systematic computations of representative microscopic foam volumes. Hereby we expect the homogenized data to be physically meaningful, e.g., symmetric and regular. The specific choice of strain-stress tuples $\boldsymbol{z}_i$ is of minor importance; they can be converted into each other using the common relations of continuums mechanics. Therefore the data sets

$$\mathcal{D}_F = \{(\boldsymbol{F}, \boldsymbol{P})_i, i = 1, \ldots, n\} \qquad \text{and} \qquad \mathcal{D}_C = \{(\boldsymbol{C}, \boldsymbol{S})_i, i = 1, \ldots, n\}$$

or its small strain equivalent $\mathcal{D}_\epsilon = \{(\boldsymbol{\epsilon}, \boldsymbol{\sigma})_i, i = 1, \ldots, n\}$ are treated equally and denoted as $\mathcal{D}$ subsequently.

To generate a data set that describes the homogenized material behavior of the foam, a deformation $\bar{\boldsymbol{F}}$ of the RVE is prescribed and the stress field is computed with a linear or non-linear FEM. The homogenized stresses $\bar{\boldsymbol{P}}$ are derived by

$$\bar{\boldsymbol{P}} = \frac{1}{|V|}\int_V \boldsymbol{P}(\boldsymbol{X})\,\mathrm{d}V\,,$$

i.e. for all integration points the stress components multiplied with the corresponding volume are summarized and divided by the RVE's volume V. The combination of the homogenized quantities $\bar{\boldsymbol{F}}$ and $\bar{\boldsymbol{P}}$ provides one data tuple. The data sets, which are used as an input for the DD-FEM, are the collections of all $(\bar{\boldsymbol{F}}, \bar{\boldsymbol{P}})$ tuples or, with $\bar{\boldsymbol{C}} = \bar{\boldsymbol{F}}^T\bar{\boldsymbol{F}}$ all $(\bar{\boldsymbol{C}}, \bar{\boldsymbol{S}})$ tuples. Since only one data point is determined per RVE simulation and these can be relatively expensive, the question arises as to how many points are required for a DD-FEM calculation. Our experience suggests a minimum number of $n = 50 - 100$ data tuples per stress component, which results in the computation of n^6 tuples for three dimensions. That many simulations can not be deduced for detailed representative

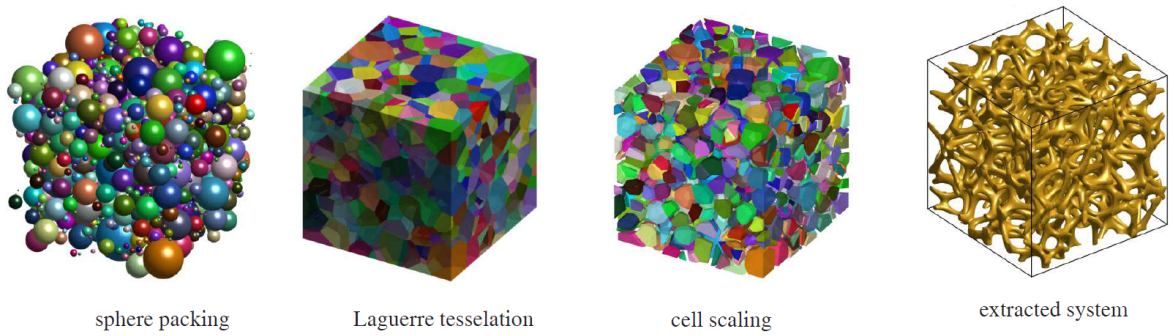


Figure 6.1: Procedure to generate a representative stochastic foam volume: sphere radius and location are determined by stochastic variables. A Laguerre tesselation assigns a domain to every sphere. Then the cells are scaled to a given volume ratio and the skeleton is extracted afterwards.

microscopic structures. Therefore, after explaining the foam RVE in more detail, we will discuss in the following how it is possible to speed up the whole process of DD-FEM calculations with RVE data. At first, we discuss how we gain data efficiently for different cases of material characteristics, and at second, we present a numerical scheme for the DD-FEM computation that needs fewer data.

6.2.1 Generation of stochastic RVEs for open-cell foam

Elastic open-cell foams are rubbery materials with a low relative density $R = \varrho_{\text{foam}}/\varrho_{\text{elastomer}} \leq 1/3$. Their mechanical behavior is determined by the matrix material and the cellular microstructure, consisting of a network of ligaments connected at junctions (vertices). An accurate description relies on the real foam's geometry, whose topological characteristics are gained from computed tomography scans for example, cf. [19]. Assuming these characteristics, e.g. the pore volume fraction, the size distribution, the coefficient of variation, and the anisotropy factor, to be known, the following procedure is chosen to generate the corresponding RVEs.

At first, random sphere distributions with a collective rearrangement of the spheres are used to build a dense isotropic packing, see the left image of Fig. 6.1. A force-biased packing algorithm [12] yields a very efficient procedure to attain the desired arrangement. At next, a Laguerre tessellation, which is a Voronoi tessellation weighted with the sphere radius [81], is used to partition the volume into subregions, see the second image of Fig. 6.1. Then, the effective density of the foam is gained by a scaling of the cells, see the third image of Fig. 6.1. Finally, the edges need to be extracted. Originally they had been shared by three cells and through the scaling they have now a hexagonal cross section (a triangular shape with cut corners). These polygonal struts are the foam's ligaments and still need to get realistic cross sections. The adjacent

struts are joined in the junctions and because this results in kinks and internal corners a smoothing algorithm is applied. We make use of a spline-based algorithm proposed in [82]. The result can be seen in the right image of Fig. 6.1.

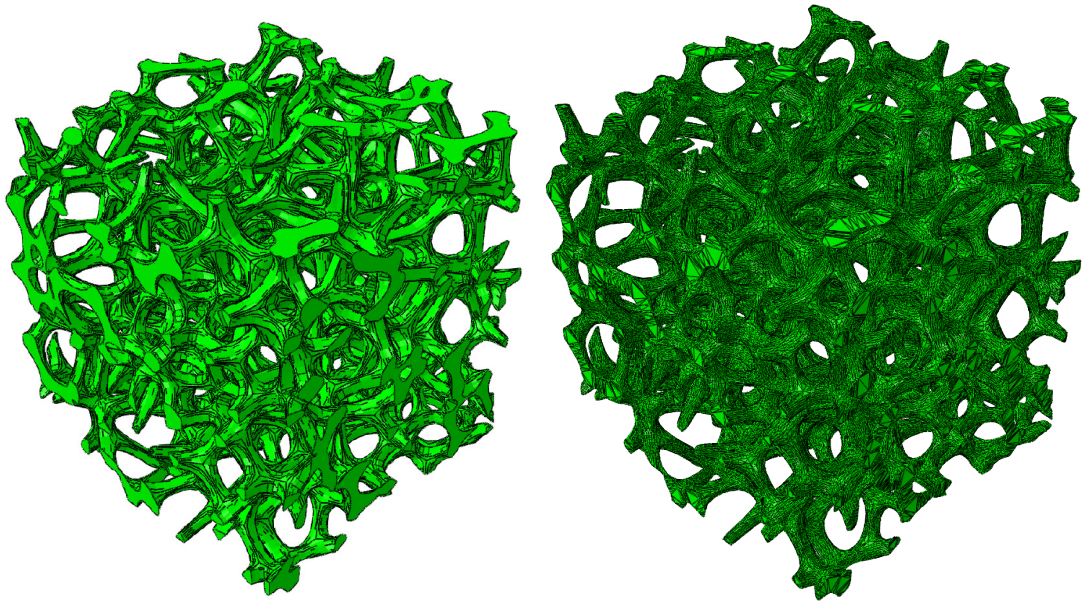


Figure 6.2: RVE of a foam with 100 log-normally distributed pores, a coefficient of variation of 0.5, and a relative density of 0.15 (left); its finite element model consists of 708 372 tetrahedral elements (right).

For a finite element analysis the resulting RVE still needs to be meshed. We employ here linear tetrahedral solid elements; an example RVE of size $1000 \times 1000 \times 1000\,\mu\text{m}$ can be seen in Fig. 6.2. The displayed RVE is meshed with 708 372 elements and a single linear computation with periodic boundary condition takes around 10 minutes. This effort is important because many – linear and non-linear – computations need to be conducted to obtain the required data sets.

6.2.2 Numerical cost reduction by using characteristic material properties

Case A: non-linear and anisotropic material
For the general situation of a direction dependent and non-linear material response, e.g. a viscoelastic foam with elongated pores in one direction, simplifications are difficult. Here only a brute force approach of one RVE computation for every $(\bar{\boldsymbol{F}}, \bar{\boldsymbol{P}})$ or $(\bar{\boldsymbol{C}}, \bar{\boldsymbol{S}})$ tuple gives the desired data set $\mathcal{D}$. Such a strategy is hardly feasible with the RVEs described above, only a simpler beam discretization may help. Additionally, it needs to be considered whether these volume elements are still representative.

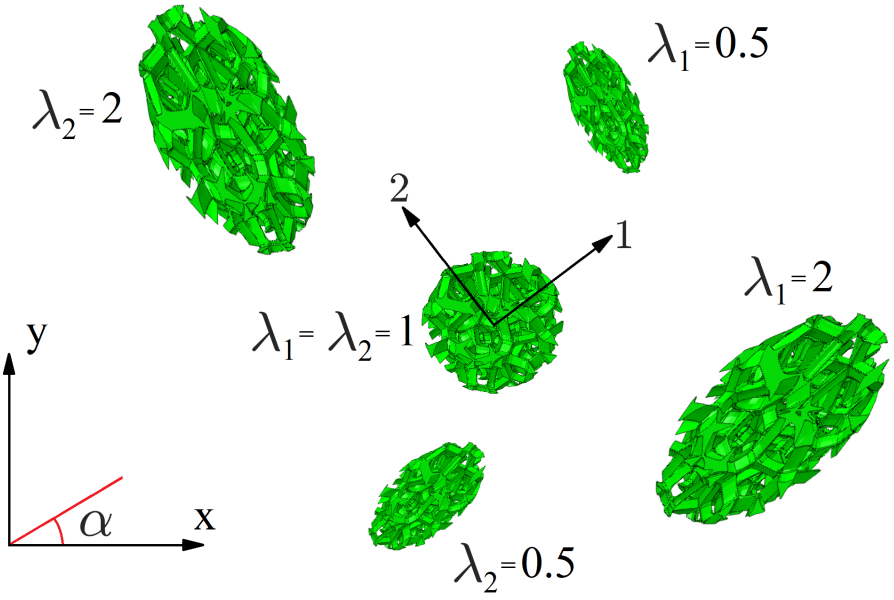


Figure 6.3: Elongation in principle stretches and rotation to the reference coordinate system for d=2

Case B: non-linear and isotropic material
For an isotropic material, i.e. when the microscopic response is non-linear but does not depend on the loading direction, we gain the possibility to rotate the RVE's coordinate system, see Fig. 6.3. Our strategy is here to sample discrete values of the principal stretches λ_α, $\alpha = 1 \dots \mathrm{d}$, and then transform the resulting principal stresses,

$$\bar{\boldsymbol{S}} = \boldsymbol{Q}\bar{\boldsymbol{S}}^{\lambda_\alpha}\boldsymbol{Q}^T$$

where $\boldsymbol{Q} \in \mathrm{SO}(3)$ is the corresponding three-dimensional rotation tensor composed of the rotations around the coordinate axes, [67]. With a stepwise rotation around these axes all deformation states can be mapped and summarized in $\mathcal{D}$. For n^6 data then n^3 computations are needed.

Case C: linear and anisotropic material
The computation eases significantly when the applied deformation is small and the microscopic response is linear. Without isotropy of the material still RVE computations with loads in every distinct direction need to be performed but the results now can be superposed

$$(\bar{\boldsymbol{C}}, \bar{\boldsymbol{S}})_k = a(\bar{\boldsymbol{C}}, \bar{\boldsymbol{S}})_i + b(\bar{\boldsymbol{C}}, \bar{\boldsymbol{S}})_j$$

with $a, b \in \mathbb{R}$. Conveniently, unit loads as displayed in Fig. 6.4 are computed and evaluated. This strategy leads to a enormous decrease of computational effort and will later be employed for the small deformation data, see A.

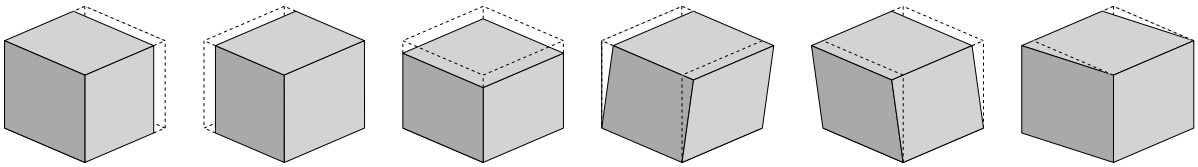


Figure 6.4: Unit load deformations of an RVE for direction dependent linear material.

Case D: linear and isotropic material
In the simplest case of isotropic, linear (elastic) material the loading scenarios simplify even more. The material's isotropy makes it possible to describe the material by only two states, namely one of the three elongations and one of the three shear states from Fig. 6.4. All other data tuples can be gained by linear combinations and rotations of the coordinate system.

Here we remark that for these simplifications we assume the RVEs to give the same averaged response, i.e., we neglect the noise induced by the stochastic generation. The uncertainty in the data will be subject of a subsequent work.

6.2.3 Numerical cost reduction by using a multi-level method

The computational effort of a data-driven multiscale analysis is enormous. As outlined above, for a non-linear microscopic material response many RVEs need to be computed to gain the required data. Additionally, data-driven computations are numerically costly due to the search for the optimal data points and, furthermore, for a macroscopic non-linear kinematic the solution requires an iterative solution procedure. Other works reduce the numerical costs by efficient nearest-neighbor searches [35] or efficient initialization of data points [78]. We apply a multi-level method, which was introduced in [75], to reduce these computational costs. In this procedure we compute the solution on smaller subsets of the original set $\mathcal{D}_l \subset \mathcal{D}$. After a computation at level l with data set $\mathcal{D}_l$, relevant data from the total set $\mathcal{D}$ are added to the next set $\mathcal{D}_{l+1}$. Here we go a step further than in [75], and use the multi-level method to identify regions where more RVE calculations are needed and then calculate additional deformation states on the fly.

Fig. 6.5 shows exemplary three data sets on a (simplified) state space. The goal is to cover the space of possible states (black square) as well as possible by data tuples (dots). Instead of a high density of data, at first only a coarse set $\mathcal{D}_0$ is created through RVE calculations (black dots, level 0). This set $\mathcal{D}_0$ is used for a first DD-FEM of the structure.

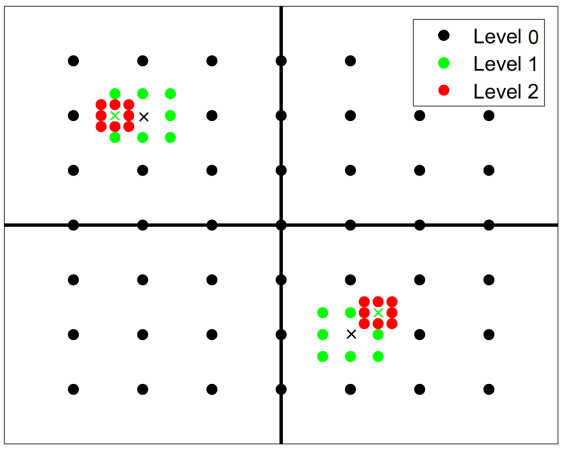


Figure 6.5: Illustration of the multi-level method for a simple state space: The black data points symbolize the input set of the initial level $\mathcal{D}_0$; the crosses are the data points actually assigned to the material points. The green and red points mark the data of the first and second refinement level, $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.

In this computation, the approximate data of relevance are identified by the data tuples which are assigned to the material points (black $\times$). Then the data set is adaptively refined and the data-driven FEM is repeated with a set $\mathcal{D}_1$. In Fig. 6.5 this is exemplarily illustrated by the two green $\times$, marking the selected data of set $\mathcal{D}_1$. Around those two assigned data tuples the data grid is refined. This leads to the data set $\mathcal{D}_2$ (red dots). This procedure can be repeated as often as needed to achieve a sufficient accuracy. The adaptive computational strategy for the multi-scale simulation is illustrated in Fig. 6.6.

6.3 Numerical Examples

To demonstrate the different DD-FEM approaches, we start our computations with a parametric study of a simple rod. Then we show the applicability of the method for the computation of engineering components by means of a polyurethane rubber sealing.

6.3.1 Example 1: Rod under tension

At first we investigate the influence of different data sets together with a linear or a non-linear kinematic. For these parametric studies we consider a one-dimensional rod

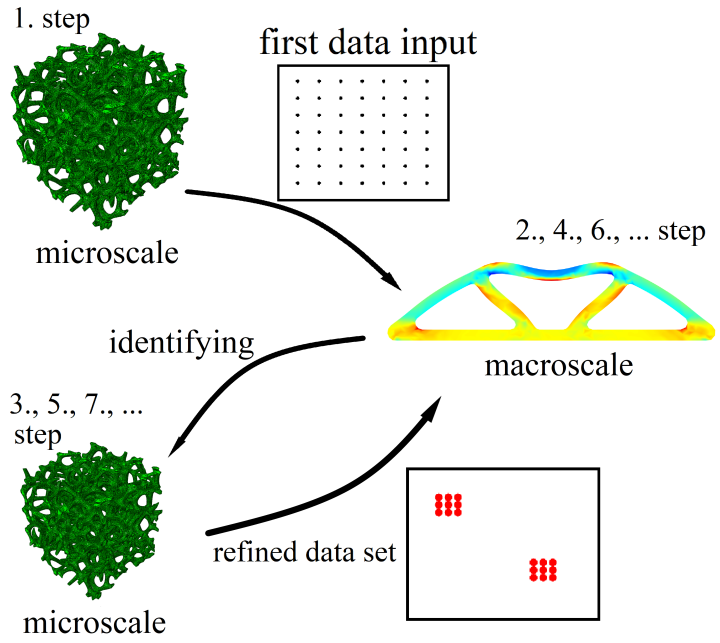


Figure 6.6: Illustration of the multi-level approach. A coarse data set is generated by RVE simulations first. A macroscopic data-driven simulation afterwards identifies region where more precise data is needed. Additional RVE simulations are conducted to generate new data in the desired regions. The latter two steps can be repeated as often as desired.

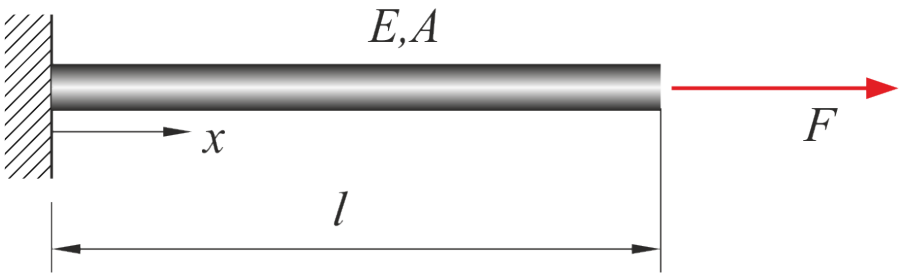


Figure 6.7: Geometry and boundary conditions of the example of Section 6.3.1: the rod is fixed on one side and a force pulling on the other side induces an homogenous tension.

of length $l = 100\,\mathrm{mm}$ and cross-section $A = 1\,\mathrm{mm}^2$ under uniaxial tension, Fig. 6.7. Assuming incompressibility of the material, the state of deformation is given by

$$\boldsymbol{F} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1^{-1/2} & 0 \\ 0 & 0 & \lambda_1^{-1/2} \end{pmatrix},$$

which means we can uniquely describe it with the lateral stretch λ_1. In this example, we do not obtain the material data sets from any particular foam RVE because this would not be incompressible. Instead we generate the data artificially. These data sets are then used in DD-FEM computations with linear and with non-linear kinematic. We remark that we refer here to the $(\boldsymbol{F}, \boldsymbol{P})$ algorithm of Section 6.1.1 as linearized, it can as well be formulated with $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$, see **??**. The non-linear kinematic follows the algorithm of Section 6.1.2 and A.

Artificial material data generation We consider the lateral stretch λ_1 of the rod to be prescribed and assign the corresponding first Piola-Kirchhoff stress P_1. Three sets of data are generated, assuming one proportional stretch-stress relation and two different strain energy densities, whereby in the latter cases we presume the continuum mechanics relation $\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P}$ to apply.

(i) Proportional (linear) data: Assuming proportionality between λ_1 and P_1 as well as λ_1^2 and S_1 we gain the following data sets:

$$\begin{aligned}(F, P)_i &= \left(\lambda_1,\, c_1(\lambda_1 - 1)\right)_i \\ (C, S)_i &= \left(\lambda_1^2,\, c_1(\lambda_1^2 - 1)\right)_i \qquad \text{for } i = 1, \ldots, n\end{aligned}$$

with material constant $c_1 = 1\,\mathrm{MPa}$.

(ii) Neo-Hooke-like data: Let the data derive from an incompressible material's strain energy density of the form

$$W = c_1(\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{F}) - 3) + p(\det \boldsymbol{F} - 1)$$

where p is a Lagrange multiplier enforcing incompressibility, we deduce the two data sets from the corresponding analytical solution, cf. [96].

$$\begin{aligned}(F, P)_i &= \left(\lambda_1,\, 2c_1(\lambda_1 - \lambda_1^{-2})\right)_i \\ (C, S)_i &= \left(\lambda_1^2,\, 2c_1(1 - \lambda_1^{-3})\right)_i \qquad \text{for } i = 1, \ldots, n\end{aligned}$$

The single material constant is here $c_1 = 1/6\,\mathrm{MPa}$.

(iii) Yeoh-like data: Using a strongly non-linear strain energy density of the form

$$W = c_1(\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{F}) - 3) + c_3(\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{F}) - 3)^3 + p(\det \boldsymbol{F} - 1)$$

we derive in the same way the data tuples, cf. [115]

$$\begin{aligned}(F, P)_i &= \left(\lambda_1,\, 2(\lambda_1 - \lambda_1^{-2})\left(c_1 + 3c_3(\lambda_1^2 + 2\lambda_1^{-1} - 3)^2\right)\right)_i \\ (C, S)_i &= \left(\lambda_1^2,\, 2(1 - \lambda_1^{-3})\left(c_1 + 3c_3(\lambda_1^2 + 2\lambda_1^{-1} - 3)^2\right)\right)_i \quad \text{for } i = 1, \ldots, n\end{aligned}$$

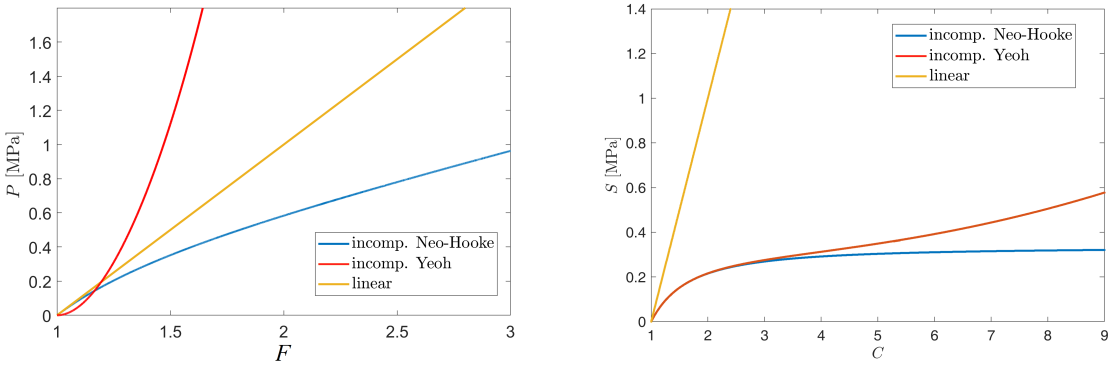


Figure 6.8: Material data of the three sets in the example of Section 6.3.1.

with constants $c_1 = 1/6$ MPa and $c_3 = 1/1000$ MPa.

The data of the three sets are plotted in Fig. 6.8. Because we chose very dense data sets with $n = 10\,000$, single data points are not recognizable in the plot but the non-linearity can clearly be seen.

Comparative study We are interested in the force-displacement relation for a homogenously elongated bar and evaluate the maximum displacement at the bars end u$= (\lambda_1 - 1)l$. The corresponding force-displacement curves are plotted for the proportional material data of set (i) in Fig. 6.9. Clearly, computing the DD-FEM with the linear kinematic gives a linear $u - f$-relation, whereas with the non-linear kinematic of Section 6.1.2 the maximum displacement is reduced. The same situation is displayed in Fig. 6.10 for the DD-FEM computation with set (ii) and in Fig. 6.11 for data set (iii).

For the sublinear material data set (ii), two effects work against each other: The non-linear data are softer for larger displacements but the non-linear kinematics stiffens the model. In other words, while the kinematic gives a higher stretch at the same displacement, the proportionality of model (ii) gives a lower stress at the same stretch. The difference between the two approaches is about 14%, 21% and 41% at $\lambda_1 = 1.5$, $\lambda_1 = 2$ and $\lambda_1 = 3$, respectively.

For the strongly non-linear material data set (iii), the stiffening non-linearity dominates the situation, Fig. 6.11. The $u - f$-curves are similar to each other. The linear or non-linear kinematic hardly makes a difference here, also because the displacements are significantly smaller than before. The relative difference between both DD-FEM computations is 8% and 19% at $\lambda_1 = 1.5$ and $\lambda_1 = 2$, respectively.

Summarizing we state that in the DD-FEM the structure of the material data set dominates the solution. For moderate straining and considering some expected noise in the

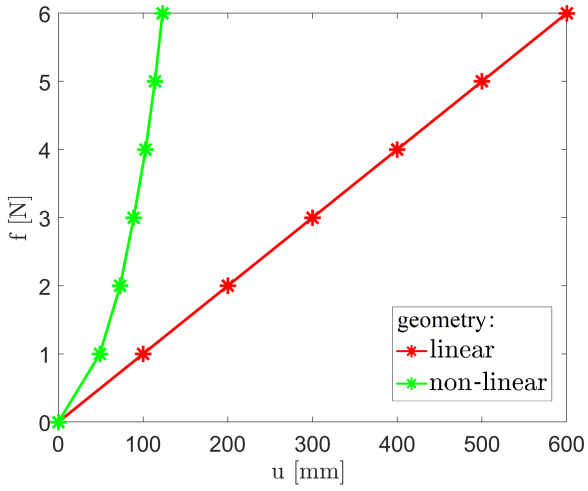


Figure 6.9: Load-displacement curves for proportional material sets (i) computed with the linear and the non-linear kinematic.

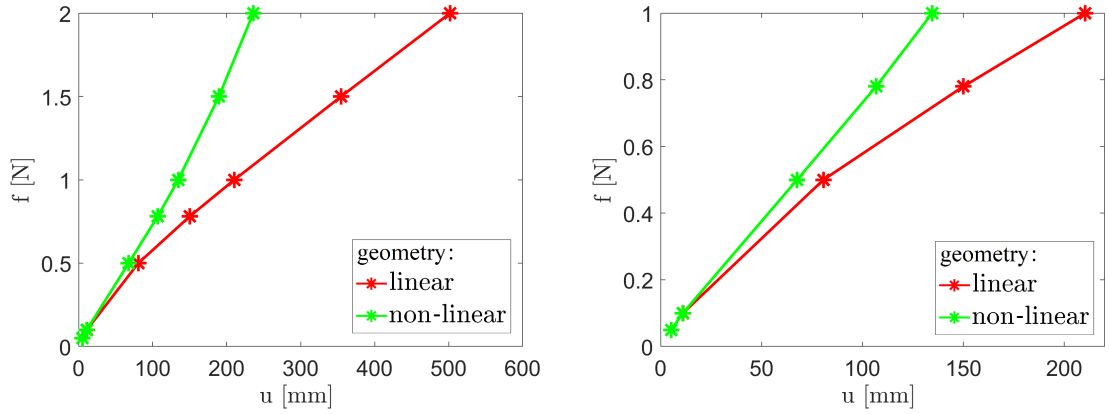


Figure 6.10: Load-displacement curves for the non-linear material set (ii) computed with the linear and the non-linear kinematic; two different regimes with 500% straining (left) and 100% straining (right) are shown.

available data, a non-linear material behavior can be mapped with a linear kinematic sufficiently accurate. This result is important because a non-linear kinematic with a Newton-Raphson iteration and a data search in every step, as described in Section 6.1.2 and A, is by several factors more expensive than a linear DD-FEM computation.

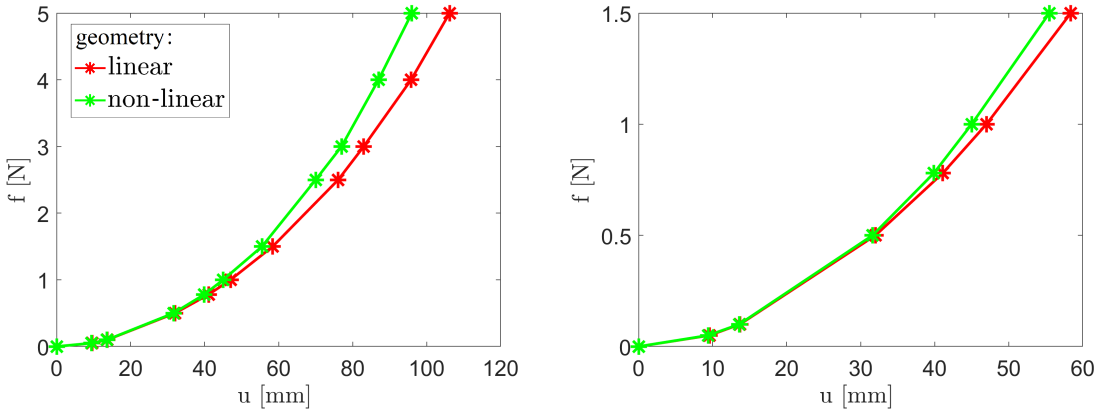


Figure 6.11: Load-displacement curves for the strongly non-linear material set (iii) computed with the linear and the non-linear kinematic; two different regimes with 100% straining (left) and 60% straining (right) are shown.

6.3.2 Example 2: A rubber sealing in a plane strain state

Now we consider a typical engineering application, namely a car door sealing made of a foamy rubber material. The computed geometry is displayed in Fig. 6.12.

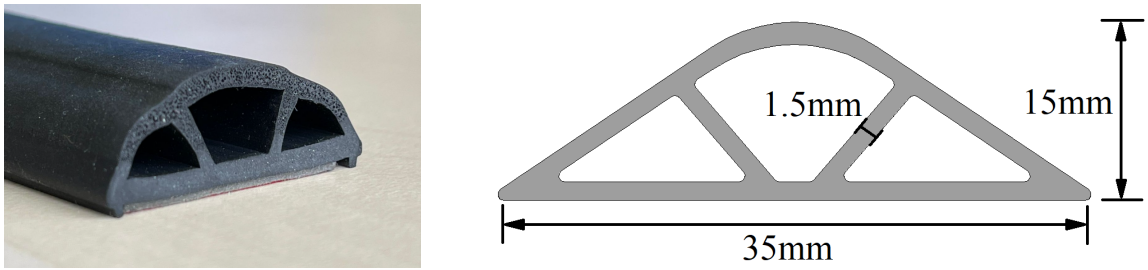


Figure 6.12: Geometry of the rubber sealing

We start with a somewhat artificial situation of a plane approximation and two-dimensional foam data. The rubber is a Neo-Hookean material extended to the compressible range, $W = c_1(\operatorname{tr} \boldsymbol{C} - 3) + {}^1\!/_{D_1}(\det \boldsymbol{F} - 1)^2$, with parameters of polyurethane, $c_1 = 3.85\,\mathrm{MPa}$ and $D_1 = 0.12\ \mathrm{MPa}^{-1}$. The computation is performed with the commercial program Abaqus.

Macroscopic finite element model The mesh of the sealing, its loading and boundary conditions are shown on left-hand side of Fig. 6.13. The component is fixed in the z-direction. For the distributed surface load p four different values are used, namely $p = 3\,\mathrm{kPa}$, $p = 10\,\mathrm{kPa}$, $p = 20\,\mathrm{kPa}$ and $p = 30\,\mathrm{kPa}$. For comparison, linear (material

and kinematic) and non-linear (material and kinematic) finite element analyses based on the classical material models were also performed. For the linear analysis homogenized parameters $E = 13\,\text{MPa}$ and $\nu = 0.4$ are used. The non-linear analysis is performed with a Neo-Hooke model and $c_1 = 7.69\,\text{MPa}$ and $D_1 = 0.08666\ \text{MPa}^{-1}$ are used.

Generation of the material data set The 2D-RVE used for data generation is a simple $7 \times 7\,\text{mm}^2$ square with 4 regularly arranged pores of radius 2 mm and meshed with 2601 finite elements that is fast and easy to compute. It is isotropical but because of the non-linear material it corresponds to Case B. Here the principal stretches λ_1 and λ_2 are varied in 51 steps between $0.8 \ldots 1.2$ which lead to 2601 non-linear finite element computations. The resulting data set has $2601 \cdot 73 = 189\,873$ data tuples which provide in our data-driven simulation the initial set $\mathcal{D}_0$ for the multi-level approach.

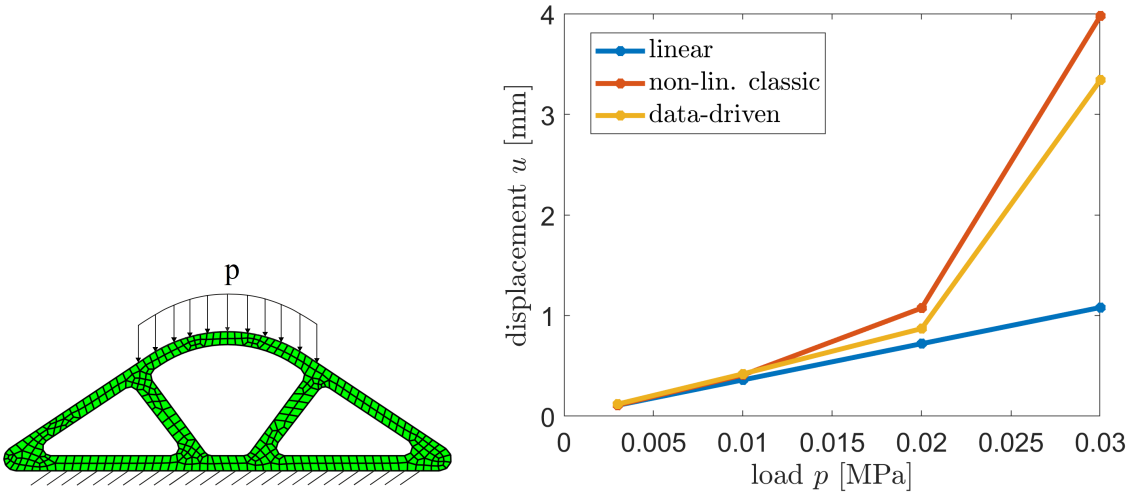


Figure 6.13: Mesh and boundary conditions of the rubber sealing on the left side. On the right side the load-displacement curve is displayed for the three simulations.

Data-driven finite element analysis The data-driven computations were run on three levels of refined data sets. In Table 6.1 the size of the input data set and the number of data tuples which are actually assigned to a material point (solution set $\mathcal{S}$) are displayed for the simulation with the highest load of $p = 30\,\text{kPa}$.

On the right-hand side of Fig. 6.13 the load-displacement curves are shown for the data-driven solution and the two classic computations. Recorded is the displacement of the center top nodes, where we have the largest displacement. The data-driven solution, computed with the non-linear material data but the linearized kinematic formulation of Section 6.1.1, gives a somewhat stiffer response than the fully non-linear Neo-Hookean

Level l	data tuples in $\mathcal{D}_l$	data tuples in $\mathcal{S}_l$
0	189.873	2.972
1	80.244	3.197
2	86.319	5.201

Table 6.1: Sizes of the input data and the number of data tuples describing the material of the multi level computation.

model. The difference in the maximal displacement is about 15% for moderate straining. As outlined in the previous example, a fully non-linear kinematic of the DD-FEM comes at the price of a much higher computational effort and so we consider the linearized data-driven response here to describe the deformation of the structure sufficiently.

In Fig. 6.14 the horizontal Cauchy stress component σ_x is displayed for $p = 20\,\text{kPa}$ and a good agreement of the classic FEM solution and the data-driven solution (level 2) can be seen.

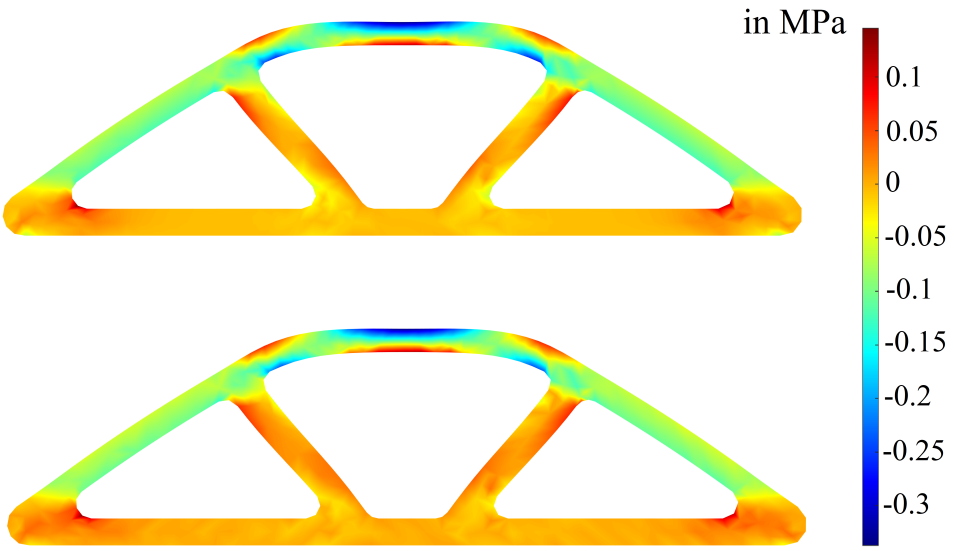


Figure 6.14: Horizontal component of Cauchy stress σ_x for a classic linear FEM solution (top) and the DD-FEM (bottom) for a surface load of $p = 20\,\text{kPa}$.

6.3.3 Example 3: Three-dimensional computation of the rubber sealing

We proceed with the rubber sealing component of Section 6.3.2 but compute now a fully three-dimensional deformation state with a different loading scenario. The material is described with microscopic RVEs of polyurethane foam.

Macroscopic finite element model The in-plane geometry and boundary conditions are the same as before, see Fig. 6.12, but the component has now a thickness of 5 mm. Additional to the surface load p_1 from atop we add two out-of-plane surface loads $\pm p_2$ pointing into the transversal z-direction and thus twisting the component, see Fig. 6.15. We set $p_1 = -7.5\,\text{kPa}$ and $p_2 = \pm 25\,\text{kPa}$.

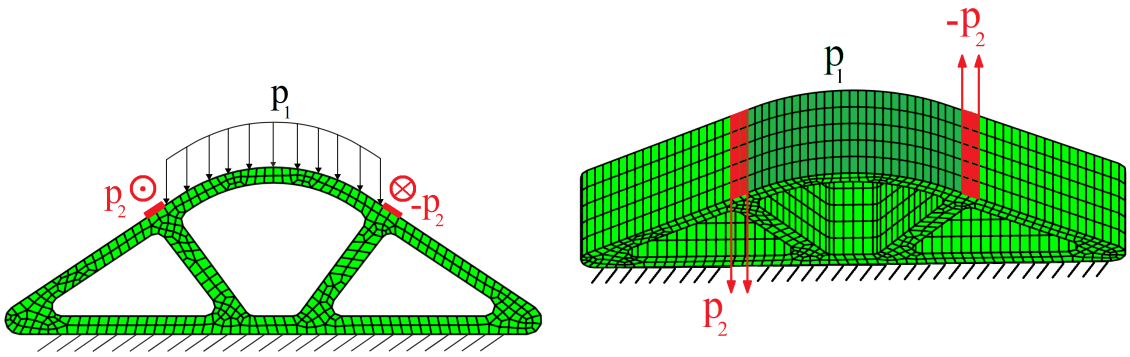


Figure 6.15: Mesh and boundary conditions of the rubber sealing in Section 6.3.3. An out-of plane surface load in z-direction is applied (red); the elements effected by the load p_1 are in darker green.

Generation of the material data set To mimic the polyurethane foam we use the RVE presented in Section 6.2.1 and compute it with the material data of Section 6.3.3. Under the loads of Fig. 6.15 we expect only moderate deformations and because of that we consider a linear behavior ($(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$ data). Due to the stochastic random sphere packing of the RVE, small deviations in the directions may still occur. Therefore, we chose the strategy outlined in Case C and conduct FEAs for the six unit loads of Fig. 6.4. The six evaluated data tuples are listed in the appendix. We observe only small deviations of around 5% in the different directions; this uncertainty is forwarded into the data set. Fig. 6.16 shows the RVE once sheared and once elongated in x direction by $\epsilon = 2\%$ and the relative low stresses. The material data set comprises stresses between $-0.2\,\text{MPa}$ and $0.2\,\text{MPa}$. The number of data tuples varies due to the multi-level approach and is listed for the different levels of refinement in Table 6.2.

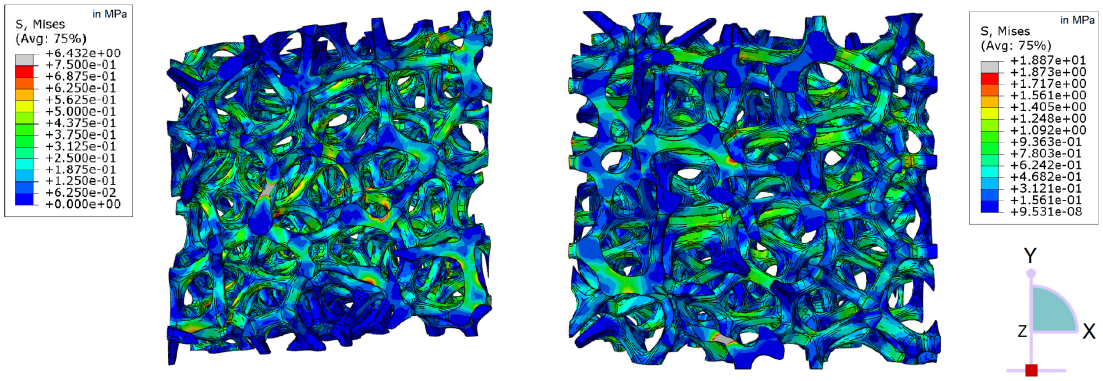


Figure 6.16: Two of the six loading scenarios of case C: On the left side a shear deformation is displayed and on the right side the RVE that is introduced in section 6.2.1 is elongated along the x-axis.

Level l	data tuples in $\mathcal{D}_l$	data tuples in $\mathcal{S}_l$	RMS
0	1.771.561	2.179	67.11%
1	1.588.491	8.889	29.17%
2	6.480.081	12.541	13.51%
3	9.142.389	14.152	9.78%
4	10.316.808	14.177	8.32%

Table 6.2: Size of the material data set $\mathcal{D}$ and number of data tuples describing the solution $\mathcal{S}$ in the multi level computation.

Data-driven finite element analysis For the data-driven computation we used the multi-level method with 4 levels of refinement. The data refinement is stopped then because the increase of additional data points is small and the root-mean-square error (RMS) is at an acceptable level, see Table 6.2. The RMS error of the stresses

$$\boldsymbol{\sigma}_{RMS} = \sqrt{\frac{\int_\Omega \Psi^*(\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{\text{ref}})\,\mathrm{d}\Omega}{\int_\Omega \Psi^*(\boldsymbol{\sigma}_{\text{ref}})\,\mathrm{d}\Omega}}$$

is calculated with respect to a standard linear finite element analysis with elastic parameters $E = 3\,\text{MPa}$ and $\nu = 0.4$, that are derived from the homogenized material. As expected the error is high of the first level computation, see Table6.2. Later, it reduces to a single-digit error which is in line with prior three-dimensional simulations. In Fig. 6.17 the horizontal stress σ_x is plotted for the reference finite element analysis and the DD-FEM computation. The displayed stress distribution shows that the results

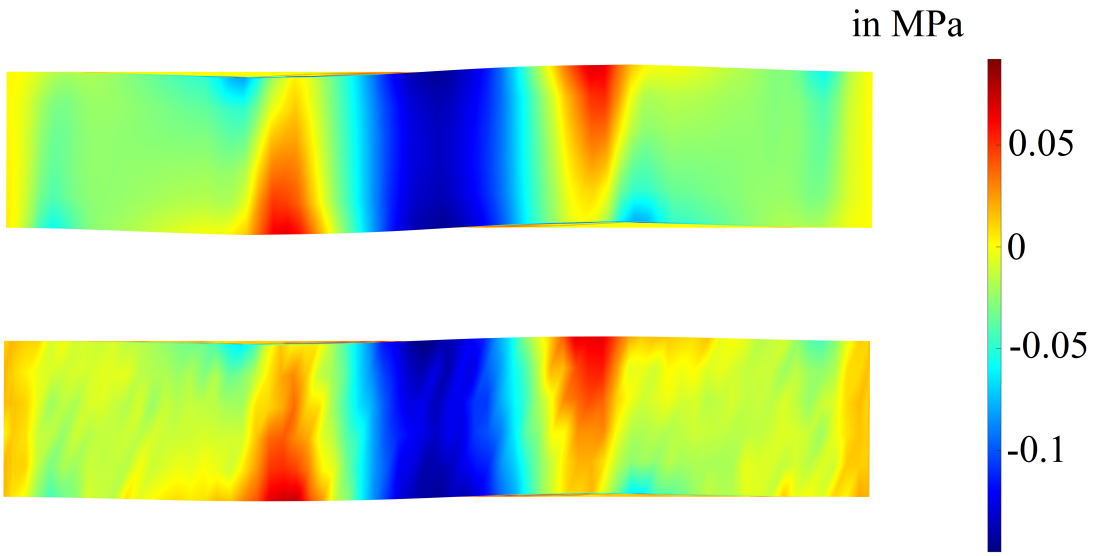


Figure 6.17: View of the normal stress σ_x from above in the rubber sealing of the DD-FEM (bottom) and the reference finite element analysis (top).

almost coincide.

The corresponding shear stress τ_{xz} is displayed in Fig. 6.18 in the deformed sealing. The displacement is magnified here by a factor of five so that the deformation of the component can nicely be seen. The numerical generation of the data can be useful for smaller stress components. In [72] it is noted that for an equidistant data grid smaller stress components could not be displayed adequately. The material density was derived by the maximal stress component and the number of data points. Since the data is generated numerically here, one would not be bound to a minimum recording resolution of an experimental setup and could still refine individual components here, which was not overused however. Finally, the data-driven method manages to perform the macroscopic FEM simulation satisfactorily.

6.4 Conclusion of the chapter

In this chapter, we propose a strategy to provide the necessary material database for linear and non-linear data-driven finite element computations. Typically, such data are presumed to be available from experimental investigations but here we suggest a computational material testing instead. With representative material volumes the micro-scale problem is investigated and the derived homogenized data give the input for the solution of the macro-scale problem.

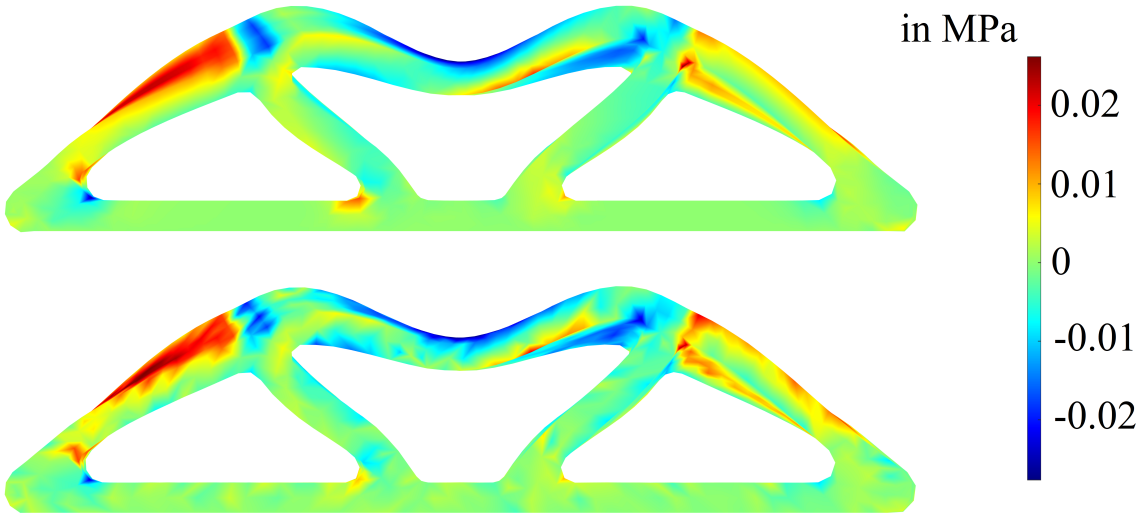


Figure 6.18: Shear stress τ_{xz} in the rubber sealing for the reference (top) and the DD-FEM (level 4). The numerical generation allows to generate suitable data in all directions.

Specifically we consider the DD-FEM of components made of open-cell foam. The representative volume elements map the foam's characteristics such as pore volume, pore distribution, pore and ligament geometry which are usually gained from CT scans. Their computation, in conjunction with simplifications made possible by the spatial material behavior, enable us to generate the material data input set.

The capabilities of the methodology are shown with three different examples. The first example is a study of how the data-driven method behaves with non-linear data and a linear setting and the other way round. In the two rubber sealing examples we focus onto the material generation and simulation of the component. From our point of view, the DD-FEM and the data-based generation of the RVE can really excel the computation of such problems. With the methods presented here, it is possible to numerically calculate the properties of the foam, and the response of a component made of it, only with the help of CT scans. No further practical or experimental action is necessary. In particular, the DD method can help in the future design of new types of foam. If a new foam is to be designed, its effect on the final product can be determined in advance by varying individual target parameters in the creation of the RVE. By selecting the target parameters in the RVE, e.g. pore volume or pore size, it is possible to determine which microscopic properties the foam has and how a component made of it would react to a given load. This whole process would be purely numerical and can save costs as well as time.

Furthermore, the data-driven method can be used to insert material properties that were not inserted from a theoretical point of view, for example non-linear data in a

linear simulation or anisotropy in an isotropic calculation.

7 Examples of applications

In this chapter, we want to investigate several small applications of the data-driven method. As already shown in Chapter 4 the data-driven FEA allows for uncertainty-specific calculations. Therefore, it can be also part of a polymorphic uncertainty computation, incorporating two or more different uncertainty models.

Furthermore, the data-driven method is applied to diffusion problems. This chapter bases on the publications [73] and [75].

7.1 Polymorphic uncertainty

Inside the priority project 1886 of the Deutsche Forschungsgemeinschaft a special focus is placed onto polymorphic uncertainty methods, i.e. the coupling of two or more models for uncertain behavior. Most of the time stochastic and fuzzy methods are used here. In our sub-project we coupled the data-driven method, as a data-stochastic uncertainty method, with a fuzzy method to gain a polymorphic uncertainty approach. Fuzzy set theory is a generalization of the classical set theory. Instead of belonging or not belonging to a set, an element gets assigned a degree of membership to the set. This degree of membership is a value between 0 and 1, where 0 means that the element does not belong to the set and 1 means the element belongs fully to the set. An introduction to fuzzy theory is found in [77] and [116]. Here we briefly introduce fuzzy variables and fuzzy numbers such that we can use them in a computation.

Fuzzy variables

In the following we focus onto one-dimensional fuzzy sets, i.e. the set of real numbers $\mathbb{R}$ can be seen as the universal set. The fuzzy set theory is a generalization of the classical crisp set formulation. In the classical formulation of set theory we distinguish only two cases of elements belonging to a set $\mathcal{X} \subset \mathbb{R}$. They can either belong to a set

$$a \in \mathcal{X}$$

or not belong to a set

$$a \notin \mathcal{X}$$

there is no in between. If we want to describe the membership of the elements by a function we can use a typical indicator function

$$\mu_{\mathcal{X}} : \mathbb{R} \to \{0, 1\}$$
$$\mu_{\mathcal{X}}(a) = 1 \text{ if and only if } a \in \mathcal{X}$$
$$\mu_{\mathcal{X}}(a) = 0 \text{ if and only if } a \notin \mathcal{X}$$

where $\mu_{\mathcal{X}}(a) = 1$ denotes that the element belongs to the set and $\mu_{\mathcal{X}}(a) = 0$ denotes the opposite. For fuzzy sets the membership value is now allowed to be a value between zero and one (including both)

$$\mu_{\mathcal{X}} : \mathbb{R} \to [0, 1]$$

This means instead of a crisp boundary we have a vague boundary. This vagueness can be used now for the description of uncertainties. Fuzzy variables are specially used for epistemic uncertainties. If a parameter can be measured, say more than a thousand times, and a distribution can be deduced, a probabilistic model yields no problems. However, some parameters cannot be measured at all or not that often to deduce probability distributions. Sometimes only a verbal statement as “ten or so” by expert knowledge is at hand. Here, it is hard to use a stochastic distribution. A discrete fuzzy set describing the above statement is shown on the left side in Fig. 7.1. The membership value on the ordinate is also called possibility. For example the value 9 got the possibility of 0.4 of being contained in the set “ten or so”. On the right side an example is given of the statement “Sarah is very young”. The possibility of a 20 year old person getting the attribute “very young” is 0.5. As fuzzy and stochastic variables express uncertainties,

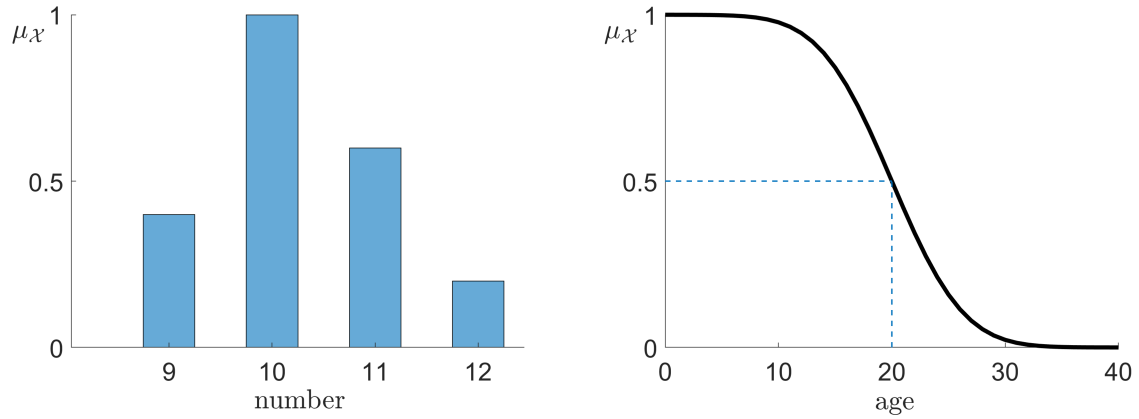


Figure 7.1: Representations of the membership functions for a discrete and a continuous fuzzy set.

have values in the range of zero and one and got other similarities both approaches are compared often. As seen before fuzzy variables can be especially useful when the amount of information is rather small. A probabilistic approach fits better when many information is at hand and a distribution can be fitted.

For uncertain input parameter which are usually a value on the real axis the special definition of a fuzzy number is used. Fuzzy numbers are fuzzy sets which have the following properties:

- defined on the real number axis
- convex
- normalized
- the membership function is piecewise continuous.

We will make use of a triangular fuzzy number as shown in Fig. 7.2. An important feature of fuzzy sets are the α-cuts of the fuzzy set

$$A_\alpha = \{a \in \mathcal{X} \,|\, \mu_\mathcal{X}(a) \geq \alpha\}$$

of which one is also displayed in Fig. 7.2. All α-cuts also define the fuzzy set completely.

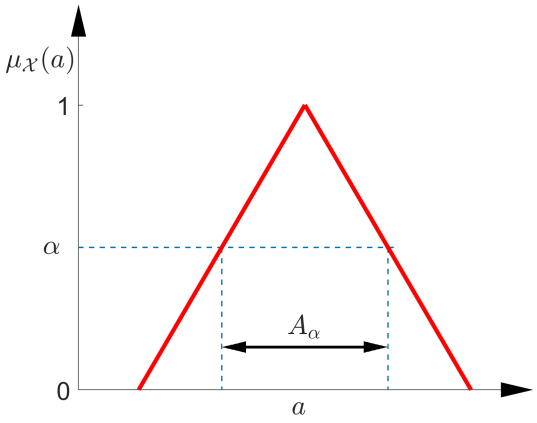


Figure 7.2: A triangular fuzzy number with an α-cut for a given α.

Fuzzy Monte-Carlo method

As already mentioned the Monte-Carlo method is a very general and practicable method to implement uncertainties. The Monte-Carlo method which we already used in Chapter 4 is now extended by a fuzzy variable. In Fig. 7.3 it is shown how to deal with

both uncertainty descriptions. The stochastic distribution is sampled as before such that we are able to compute with a deterministic value. The fuzzy variable is discretized into α-cuts which can be used for an interval propagation then. As we sample over the stochastic and the fuzzy domain there are two loops now. Many computations have to be conducted therefore. The result is a family of distribution function for the different level of assumption.

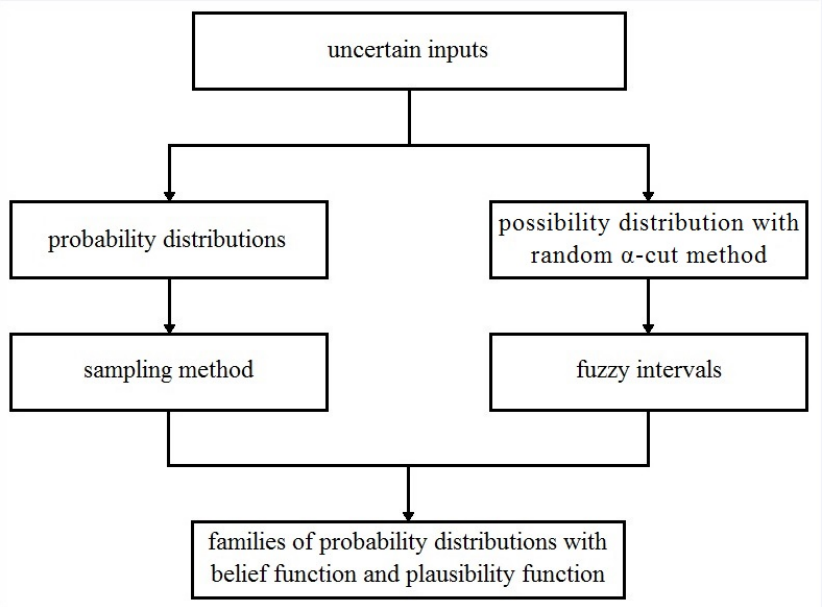


Figure 7.3: Different uncertain inputs to the Monte-Carlo method and how it is dealt with both.

Numerical example

To conduct such a computation we go back to the cantilever plate example in plane stress conditions. For the geometric properties and discretization review Section 3.4. The material is described by 25 data sets which are sampled with a mean of $3 \cdot 10^6$ Pa for the data-driven method. The force is a fuzzy triangle number which is characterized by the three points of $(58, 63, 68)$ Pa.

By the double discretization we gain a family of probability distributions for presumption levels of the fuzzy value as an output. The lower limit of the distribution functions seen in Fig. 7.4 is called belief function, the upper limit is called plausibility function. Both are displayed as solid lines on the left side of Fig. 7.4. The dashed lines are the ones with presumption level 1 and therefore correspond to a normal Monte-Carlo simulation. Overall, the results coincide pretty much. However, the cyan line of the approach where the whole beam stiffness is varied by only one stochastic variable varies more to the end.

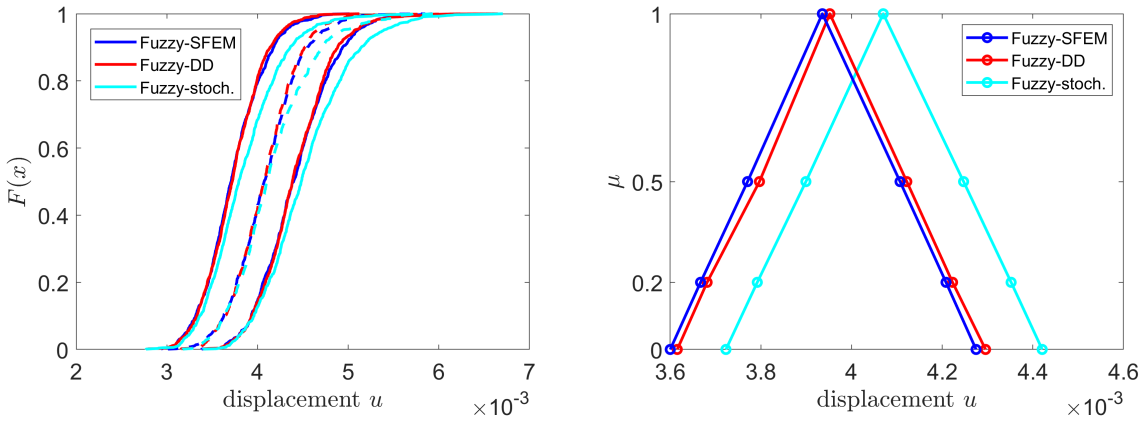


Figure 7.4: Display of the resulting family of distribution functions for the different levels of presumption (left) and the propagated fuzzy structure of the displacement.

The reason behind this is that one variable leads to way more variation. if the realization of the random variable is very small the whole beam loses stiffness. In contrast, if in the stochastic finite element method or in the data-driven approach a random variable only describes one element such that the stiffness is more depending on the average of the whole beam.

Even though the implementation of the data-driven method into a polymorphic approach is in the foreground here we have a look onto the uncertainty propagation of the fuzzy variable. On the right-hand side the profile of the fuzziness is shown for a distribution function value of 0.4. Due to the linear finite element behavior the structure of the epistemic uncertainty stays the same here. Also the probabilistic structure does not yield any new information. The still normal distributed structure can be seen in the empiric distribution functions. As already discussed the uncertainty propagation plays a smaller part in this example it shows that data-driven fuzzy computations are feasible.

7.2 Diffusion

Whereas the data-driven method of [67] has been formulated in the context of computational mechanics and has already received several enhancements and applications [59, 69, 72, 88, 34], we believe that its range and scope is much larger. Specifically we apply the data-driven finite element framework here to diffusion problems. Diffusion constitutes typical transport equations which describe numerous phenomena such as the migration of particles, electric field and temperature evolution or the spreading of diseases. All these phenomena may lead to large fields of unstructured experimental

data. Because modeling empiricism adds error and uncertainty to the solutions, the data-driven computing is expected to be particulary beneficial. Uncertainties find their way into the simulation as scattering of the data in one data set or in form of multiple data sets measured on multiple experiments.

Governing equations

The diffusion of particles fulfills the continuity equation

$$\dot{u} = f(u) - \nabla \cdot \boldsymbol{j} \tag{7.1}$$

for a particle source $f(u)$ and a flux vector $\boldsymbol{j}$. In other words, Eq. (7.1) follows from the universal principle of mass balance (or mass conservation if we disregard the source term $f(u)$ for a moment) and is, therefore, model-free. The flux, however, is usually derived from the assumption of a uniform and spatially homogeneous material. With a diffusion coefficient D it is usually modeled as $\boldsymbol{j} = -D\nabla u$ and is, at best, empirical. In the proposed data-driven approach, the flux modeling will be replaced by a time-dependent data set $\mathcal{D}_t$, which is composed of n measured values for the spatial flux vector $\boldsymbol{j}$ and the descent of particle concentration $\boldsymbol{i} = -\nabla u$

$$\mathcal{D}_t = \{(\boldsymbol{i}, \boldsymbol{j})_l^t\}_{l=1}^n \,. \tag{7.2}$$

Spatial heterogeneity can be accounted for by multiple data sets assigned to different locations. In total, the resulting data-driven problem consists of the minimization of a distance function to the current data set subject to continuity and kinematic constraints.

Specifically we illustrate the character of the data-driven boundary-value problem by recourse to the stationary solution of Eq. (7.1). It leads to a classical Poisson problem for $u(x,t)$,

$$\nabla \cdot \boldsymbol{j} = f, \qquad \boldsymbol{j} = \boldsymbol{j}(\boldsymbol{i}), \qquad \boldsymbol{i} = -\nabla u \tag{7.3}$$

where the first equation describes equilibrium (conservation of linear momentum), the second equation plays the role of a constitutive equation and the last relation is the kinematic constraint. Figure 7.5 (left) illustrates that the corresponding Poisson problem has a solution at the intercept of the two $\boldsymbol{i} - \boldsymbol{j}$-curves. As in the mechanical data-driven problem the constitutive relation is replaced by the data set $\mathcal{D}_t$ and we cannot presume that we can find an intercept with the equilibrium equation $(7.3)_1$. The aim is therefore to find the minimal distance between the physical equilibrium $\mathcal{E}$, including all constraints of the system, and the constitutive data set $\mathcal{D}$,

$$(\boldsymbol{i}^*, \boldsymbol{j}^*) = \underset{\mathcal{D}}{\arg} \left\{ \min_{\boldsymbol{x} \in \mathcal{E}} \min_{\boldsymbol{y} \in \mathcal{D}} \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathbb{D}} \right\} \tag{7.4}$$

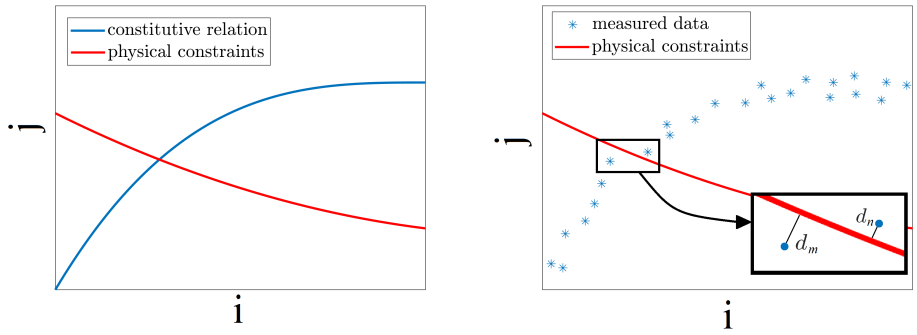


Figure 7.5: In a classical Poisson problem formulation the solution is the intercept of the equilibrium equation and the constitutive relation (left); in the data-driven framework there is not necessarily an intercept (right) and the solution needs to be approximated.

where we understand the data points which do so as the solution of the data-driven problem, $(\boldsymbol{i}^*, \boldsymbol{j}^*)$. The norm $\|\cdot\|_{\mathbb{D}}$ is defined as

$$\|(\boldsymbol{i}, \boldsymbol{j})\|_{\mathbb{D}} = \frac{1}{2} \int_{\Omega} \boldsymbol{i} : \mathbb{D} : \boldsymbol{i} + \boldsymbol{j} : (\mathbb{D})^{-1} : \boldsymbol{j} \, \mathrm{d}\Omega. \tag{7.5}$$

Norm (7.5) weights the different magnitudes (and units) of $\boldsymbol{i}$ and $\boldsymbol{j}$ by means of a numerical diffusion tensor $\mathbb{D} = D\mathbb{I}$. We remark that this tensor can be chosen arbitrarily; a non-isotropic form is also possible. Correspondingly, we define a global penalty function for the distance between numerical values $(\boldsymbol{i}, \boldsymbol{j})$ and data set test values $(\boldsymbol{i}', \boldsymbol{j}') \in \mathcal{D}$,

$$W^{\mathrm{py}} = \frac{1}{2} \int_{\Omega} [(\boldsymbol{i} - \boldsymbol{i}')^T \mathbb{D} (\boldsymbol{i} - \boldsymbol{i}') + (\boldsymbol{j} - \boldsymbol{j}')^T \mathbb{D}^{-1} (\boldsymbol{j} - \boldsymbol{j}')] \, \mathrm{d}\Omega \, . \tag{7.6}$$

Functional (7.6) needs to be minimized, subjected to the physical constraints $(7.3)_1$ and $(7.3)_3$ which are enforced by Lagrangian multipliers.

For a transient problem the temporal evolution of field $u(x, t)$ follows Eq. (7.1).

Finite element discretization

The diffusion equation is solved in weak form and integrated in time using a backward Euler method. Then, at a time step $t_k \in \{t_0 = 0, t_1, t_2, \ldots, t_{k-1} + \Delta t = t_k, \ldots\}$ the

constrained penalty functional (7.6) has the form

$$W^* = \frac{1}{2}\int_\Omega [(\boldsymbol{i}^k - \boldsymbol{i}'^k)\mathbb{D}(\boldsymbol{i}^k - \boldsymbol{i}'^k) + (\boldsymbol{j}^k - \boldsymbol{j}'^k)\mathbb{D}^{-1}(\boldsymbol{j}^k - \boldsymbol{j}'^k)\,\mathrm{d}\Omega + \int_\Omega \boldsymbol{\lambda}^k \left(\frac{u^k - u^{k-1}}{\Delta t} + \nabla\cdot\boldsymbol{j}^k - f^k\right)\mathrm{d}\Omega + \int_{\Gamma_N} \bar{\boldsymbol{\lambda}}^k(\boldsymbol{j}^k\cdot\boldsymbol{n} - \bar{\boldsymbol{j}}^k)\,\mathrm{d}\Gamma \tag{7.7}$$

We remark that the kinematic constraint $(7.3)_3$ is implicitly fulfilled by a conform finite element discretization and so only condition $(7.3)_1$ needs to be enforced by Lagrangian multipliers $\boldsymbol{\lambda}$.

For spatial discretization a standard finite element method is used. Taking the variations and with the usual calculus we arrive at the finite element system

$$\begin{aligned}
\frac{1}{\Delta t}\int_\Omega \mathbf{N}^T\mathbf{N}\,\mathrm{d}\Omega\,\boldsymbol{\lambda}^k + \int_\Omega \mathbf{B}^T\mathbb{D}\mathbf{B}\,\mathrm{d}\Omega\,\mathbf{u}^k &= \int_\Omega \mathbf{B}^T\mathbb{D}\mathbf{i}^{*k}\,\mathrm{d}\Omega \\
\frac{1}{\Delta t}\int_\Omega \mathbf{N}^T\mathbf{N}\,\mathrm{d}\Omega\,\mathbf{u}^k - \int_\Omega \mathbf{B}^T\mathbb{D}\mathbf{B}\,\mathrm{d}\Omega\,\boldsymbol{\lambda}^k &= \int_\Omega \mathbf{N}^T\mathbf{f}^k\,\mathrm{d}\Omega + \frac{1}{\Delta t}\int_\Omega \mathbf{N}^T\mathbf{u}^{k-1}\,\mathrm{d}\Omega + \int_\Omega \mathbf{B}^T\mathbf{j}^{*k}\,\mathrm{d}\Omega \\
&\quad + \int_{\Gamma_N} \mathbf{N}^T\bar{\mathbf{j}}^k\,\mathrm{d}\Gamma
\end{aligned} \tag{7.8}$$

where $\mathbf{N}$ is the matrix of shape functions, $\mathbf{B}$ the matrix of their spatial derivatives, $\mathbf{f}$ is the source vector, $\bar{\mathbf{j}}$ are prescribed boundary fluxes, and $\mathbf{u}$, $\boldsymbol{\lambda}$ are the vectors of unknowns.

To solve for the optimal data points $(\mathbf{i}^{*k}, \mathbf{j}^{*k})$ an iterative algorithm is used which starts with a random initialization or the value of the last time step $(\mathbf{i}^{*k-1}, \mathbf{j}^{*k-1})$. The finite element equations (7.8) need to be solved then. Afterwards new data points $(\mathbf{i}'^k, \mathbf{j}'^k)$ can be assigned which are used in the next iterative step of the algorithm. The algorithm stops if the change in the global penalty function is smaller than a certain threshold, i. e. when $(\mathbf{i}^{*k}, \mathbf{j}^{*k})$ is reached. More details can be found in [72, 73].

Numerical examples

Here we perform numerical simulations on a two-dimensional domain $\Omega \in [0, 100]^2$. The finite element mesh is regular and consists of $2 \cdot 25^2$ linear triangular elements (P1 elements). The boundaries are free, i.e. the flux $\bar{\boldsymbol{j}}$ is zero. In all simulations the initial state is a stamp-like cluster of concentration in the center of the domain, see Fig. 7.6 (left).

Prerequisite to a data-driven simulation are data sets, Eq. (7.4). Here the data are artificially sampled and dimensionless. The descent data space covers a square equidistantly,

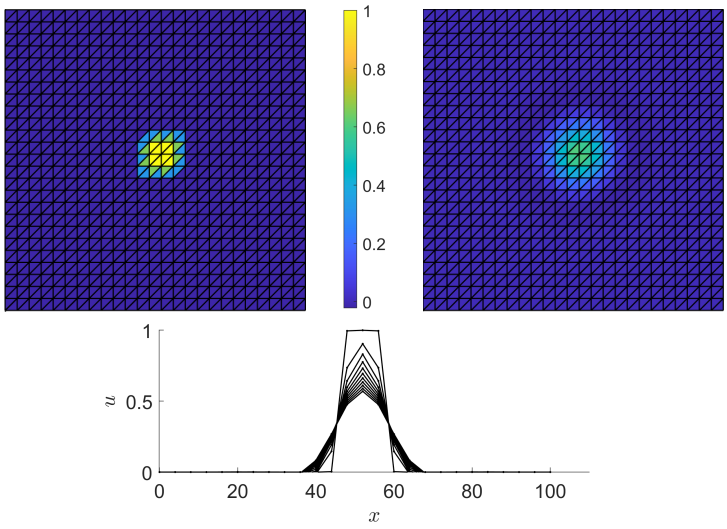


Figure 7.6: Finite element mesh with initial field u (left) and isotropic diffusion after 1000 time steps computed with data set $\mathcal{D}_1$ (middle) and concentration profile along the middle axis for every 100 time steps (right).

$\boldsymbol{i} \in [0, 10]^2$. The corresponding flux values are generated using artificial diffusion tensors $\mathbb{D}^{\text{art}}$. One data set then comprises the data points $\mathcal{D} = \{(\mathbf{i}, \mathbf{j})_l\}_{l=1}^n$ with $n = 100^2$ entries.

A first data set $\mathcal{D}_1$ is based on a simple diagonal diffusion tensor

$$\mathbb{D}_1^{\text{art}} = D_0 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{7.9}$$

and recreates isotropic diffusion with $D_0 = 0.04$. To represent real-life data, we add for the second data set $\mathcal{D}_2$ an uncertainty term. The artificial diffusion tensor $\mathbb{D}_2^{\text{art}}$ is afflicted by two stochastic terms N and U

$$\mathbb{D}_2^{\text{art}} = (1 + N) \cdot D + U \tag{7.10}$$

where both stochastic variables are 2×2 matrices

$$N = \begin{pmatrix} N_1 & 0 \\ 0 & N_2 \end{pmatrix} \qquad U = \begin{pmatrix} 0 & U_1 \\ 0 & 0 \end{pmatrix}$$

with $N_1, N_2 \sim \mathcal{N}_{0,0.1}$ and $U_1 \sim \mathcal{U}(0, 0.016)$. In this way, the diffusion is still almost isotropic but, as it is observed in common experiments, the resulting flux data scatter in a normally distributed manner. Also, by U there is put a certain weight onto the off-diagonal element which results in a small bias of the flow.

In our first simulation with data set $\mathcal{D}_1$ we show that the data-driven solution approx-

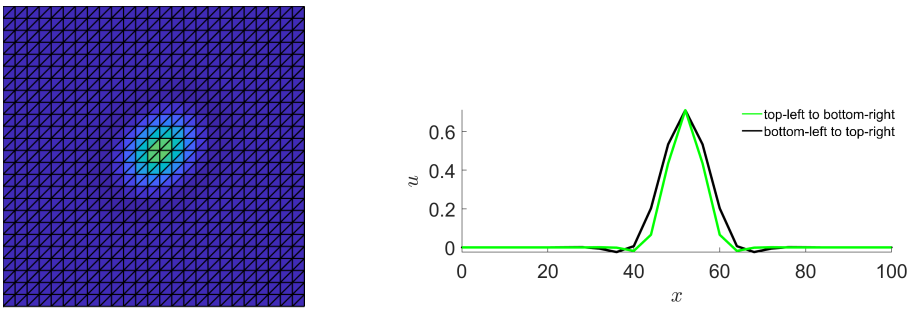


Figure 7.7: Concentration computed with data-set $\mathcal{D}_2$ after 1000 time steps of diffusion (left) and profile along the diagonals (right); the color coding refers to the legend of Fig. 7.6.

imates a classic diffusion problem. Using 100 data points per dimension we end up computing results which are close to the expected solution. In Fig. 7.6 (right) a profile through the center is shown for different time steps. Due to the sampling with an equidistant grid there is symmetry of the profile.

In a second simulation with data set $\mathcal{D}_2$ we observe the effect of the stochastic terms. A snapshot after 1000 time steps of evolution is shown in the left of Fig. 7.7. On the right of Fig. 7.7 the concentration profiles along the domain's diagonals are displayed and we can see that the concentration profile of the lower-left to the upper-right corner is wider spread than the one from top-left to bottom-right. This variation is caused by the direction-dependent data.

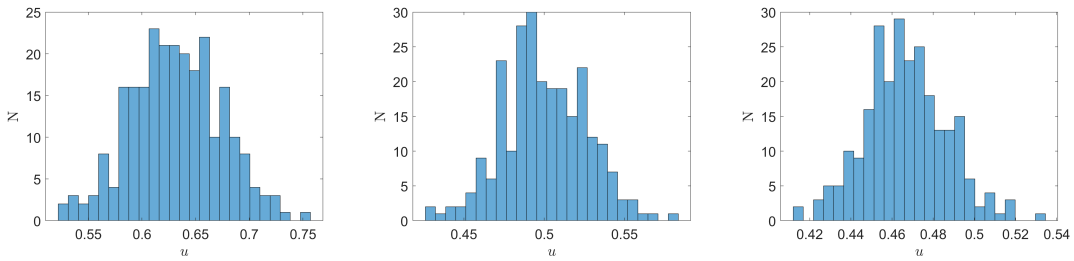


Figure 7.8: Histograms of the concentration at the center of the cluster (left), above the center (mid) and diagonally from the center (right).

Since we can not make any statement regarding uncertainties from only one simulation, we consider now the scenario of multiple measurements. Let us assume that several data sets have been measured which all have the deviations as modeled by Eq. (7.10). This results here in 200 different data sets of type $\mathcal{D}_2$ which are now used for the data-driven finite element analysis. We use one of those $\mathcal{D}_2$ data sets for the complete domain Ω but remark that it would be also possible to use different sets for subdomains. Such assumptions, however, would need a physical justification. Here we assume uncertain

data of type (7.10) for the full domain and show the resulting histograms of concentration value distribution for three different points in Fig. 7.8.

8 Conclusions

At the end of this thesis we want to summarize the results and attempt an outlook of what is next for the data-driven finite element method. After we introduced the DD-FEM approach we extensively investigated the approach and extended it in several meanings. The data-driven approach allows to circumvent the empirical fitting of material models to material data and to incorporate the data directly in finite element computations. This methodology reduces the efforts spent on the fitting process and model uncertainties completely.

We first investigated the results of the DD-FEM in a cantilever plate problem with different sizes of data sets. The data density, i.e. the area covered by data points divided by the number of data points, is a crucial input for the quality of the data-driven method. However, too much data increases the numerical cost significantly and make the computation unpractical. A number of around 100 data points per strain/stress coefficient showed good results when the covered space is between the maximal strain/stress values and the double maximal value. We then investigated the robustness of the DD-FEM towards the input of the numerical material tensor. The DD-FEM showed exceptional results. While the error is relatively small even for the halved or doubled Young's modulus the data-driven computation also gives a hint into which direction the perfect choice of value lies. At the end of Chapter 4 we compared the DD-FEM to the stochastic finite element method to incorporate material uncertainties. Those can be included in the DD-FEM in a natural way as we don't need to choose a model here too. The material uncertainties are implemented by multiple data sets which correspond to the measurements. The DD-FEM reproduced the resulting distribution of the SFEM successfully with a small shift due to the discrete data set. A mathematical convergence can be assumed too which, however, was not proven here.

One drawback of the data-driven approach is the increase of the numerical cost. While the effort of solving the second system of equations is low due to the same stiffness matrix, the numerical cost of the nearest-neighbor search increase the cost rapidly. Here we proposed a numerical method that splits the data-driven computation into more simulations with an adapting data set. In the first layer a coarse data set is used to only identify the regions of material states needed in the simulation. The following layers increase the precision by refined data set in the identified area. With this method, we are able to conduct full three-dimensional data-driven computations with a relatively small computing time and storage demand.

Most data sets used in this thesis are synthetic data sets sampled with a hidden material law. In Chapter 6 we generated a data set from RVE calculations and used them for DD-FEM computations. Performing measurements in three dimension to deduce the material behavior are still hard and expensive to perform. Therefore, a numerical alternative is presented here. The foundation of this approach are μCT-scans of the material microstructure. A representative volume element is generated with the deduced information. Simple deformation states are used to generate the basis of the data set. Due to the linear regime of loading a large database can be generated afterwards and used in the DD-FEM simulations.

In Chapter 7 two applications of the data-driven method are briefly investigated to illustrate the possibilities of this method. First, the DD-FEM is used in a polymorphic uncertainty computation together with an additional fuzzy variable. From the results fuzzy-probability distributions can be deduced. The data-driven approach is an interesting ansatz for polymorphic descriptions as the variable which is described by the data does not need to be sampled. The data that is at hand is a realization of the uncertain parameter value.
In the second part of the chapter the data-driven framework is applied to a diffusion problem. It shows the easiness of applying the DD-FEM to a different problem. Again probability distributions are computed where this time the stochastic part influences the isotropy of the diffusion process.

Outlook

Many data-driven methods have been presented in the recent past. These are often used for the material description in mechanical engineering. They allow the use of material data without explicitly employing an analytical function, a so-called model-free simulation. While this can be useful in general when we have problems fitting the data to known models, it does not make sense when the data can be fitted to a known model. A physically motivated and interpretable model allows much more insight into the physical processes and so the attempt to replace constitutive laws also meets criticism. David Gonzalez comments in his paper [46]: “While there is a growing interest in this sense around the machine learning community, some recent works have attempted to simply substitute physical laws by data. We believe that getting rid of centuries of scientific knowledge is simply nonsense. There are models whose validity and usefulness is out of any doubt, so try to substitute them by data seems to be a waste of knowledge. While it is true that fitting well-known physical laws to experimental data is sometimes a painful process, a good theory continues to be practical and provide useful insights to interpret the phenomena taking place”. This criticism must be agreed under the above points, which is why the future of these methodologies definitely lies in the areas of new and complex materials for which current material models must be adapted again and

again. For example, foam-like structures or the currently popular additive manufacturing provides new and different material behaviors. Finding a separate material law for each new printing pattern would be a great effort. This is where data-driven methodologies can stand out and present fast and attractive solutions.

The future of the data-driven computational analysis

The data-driven methodology presented in this thesis is still very young. It attempts to form further foundations for the development of the method. The DD-FEM does not only replace the material modeling process, but even removes it. Instead of a material model, the data itself is directly used in the finite element computation. Up to this point, the method shows satisfactory results. Model risks are avoided, and with a sufficient data density, the solution is equivalent to those of the classical FEM with a material model. Additional numerical effort can be kept low in the future by means of intelligent search and initialization algorithms.

As a still new approach, simpler models are used in the DD-FEM currently. As mentioned above, these will certainly have to be discarded in the future. From a theoretical point of view, the method has already been extended to further applications like inelasticity, diffusion, fracture and some more. One of the future goals must be the practical application, that is, measuring data and using it in simulations. This is already very difficult and costly for recording three-dimensional strain-stress data. To record such data for diffusion processes or crack growth seems to go one step further, so the practical applicability must not be lost sight of. If the data-driven method is to prevail in the future, the structured construction of a material database must be simple and feasible. If it is even feasible to collect such data sets in-situ, the data-driven methodology can help create so-called digital twins. Another way could be the numerical generation of data as presented in this thesis. Therefore, the possibility to produce data with models of the micro structures should also be further researched.

However, there is also still room for more theoretical research. The directions in which we see further research are divided into three areas: the method itself, data manipulation, and the further expansion of applications. First, the energy densities that were chosen so far, together with the numerical stiffness parameter, still need to be improved, especially for nonlinear data. A fixed weighting constant leads to converging problems that need to be further researched. Second, the data used in this thesis is always unprocessed to implement the uncertainties as they were measured. However, interesting approaches using cluster theory or tensor voting manipulate the data for faster computation or more insight. Many statistical methods may be used upon the data before the computation. Third, also further application areas can still be researched such as electrostatics, hydrodynamics or others where a constitutive equation plays a major role and measurements may be easier for three-dimensional data.

All in all, the data-driven methodology seems to be a promising method that is worth researching. The knowledge about the possibility to use the data itself should solely motivate users to generate, record and save more data. Then, the data-driven method can excel with the growth of big data repositories of material data in the future.

List of Figures

List of Tables

Bibliography

[1] Forbes; A Very Short History Of Data Science. https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/#503b3c5955cf. Accessed: 2019-12-11.

[2] Forbes; data science: What's the half-life of a buzzword? https://www.forbes.com/sites/gilpress/2013/08/19/data-science-whats-the-half-life-of-a-buzzword/#16090b907bfd. Accessed: 2019-12-11.

[3] Github: uncertainty-toolbox. https://github.com/uncertainty-toolbox/uncertainty-toolbox/blob/master/docs/images/logo.svg. Accessed: 2019-11-16.

[4] Learning data science in 6 weeks - how you can do it? https://hackernoon.com/learning-data-science-in-6-weeks-how-you-can-do-it-d46520c12d43. Accessed: 2019-12-11.

[5] *Matlab Documentation*. Natick, MA, 2005.

[6] *Abaqus User Manual*. RI, USA, 2014.

[7] A. Abdulle, E. Weinan, B. Engquist, and E. Vanden-Eijnden. The heterogeneous multiscale method. *Acta Numerica*, 21:1–87, 2012.

[8] G. Allaire. Homogenization and two-scale convergence. *SIAM Journal on Mathematical Analysis*, 23(6):1482–1518, 1992.

[9] D. Anders, A. Hoffmann, H.-P. Scheffler, and K. Weinberg. Application of operator-scaling anisotropic random fields to binary mixtures. *Philosophical Magazine*, 91(29):3766–3792, 2011.

[10] J. D. Arregui-Mena, L. Margetts, and P. M. Mummery. Practical application of the stochastic finite element method. *Archives of Computational Methods in Engineering*, 23(1):171–190, 2016.

[11] J. D. Arregui-Mena, L. Margetts, and P. M. Mummery. Practical application of the stochastic finite element method. *Archives of Computational Methods in Engineering*, 23(1):171–190, 2016.

[12] M. Bargieł and J. Mościński. C-language program for the irregular close packing of hard spheres. *Computer Physics Communications*, 64(1):183–192, 1991.

[13] H. H. Bauschke and J. M. Borwein. On the convergence of von neumann's alternating projection algorithm for two sets. *Set-Valued Analysis*, 1(2):185–212, 1993.

[14] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM review*, 38(3):367–426, 1996.

[15] O. Bayer. Das di-isocyanat-polyadditionsverfahren (polyurethane). *Angewandte Chemie*, 59(9):257–272, 1947.

[16] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[17] J. Biehler and W. Wall. The impact of personalized probabilistic wall thickness models on peak wall stress in abdominal aortic aneurysms. *International journal for numerical methods in biomedical engineering*, 34(2):e2922, 2018.

[18] F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, and B. Klusemann. A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials*, 6:110, 2019.

[19] S. Buchen, L. Bogunia, S. Siebert, and K. Weinberg. Microstructure characterization of foamed rubber based on intrinsic volumes of μct-images. *in preperation*, 2022.

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

[21] D. C. Charmpis, G. I. Schuëller, and M. Pellissetti. The need for linking micromechanics of materials with stochastic finite elements: A challenge for materials science. *Computational Materials Science*, 41(1):27–37, 2007.

[22] F. Chinesta, P. Ladeveze, R. Ibanez, J. V. Aguado, E. Abisset-Chavanne, and E. Cueto. Data-driven computational plasticity. *Procedia engineering*, 207:209–214, 2017.

[23] D. Cioranescu, A. Damlamian, and G. Griso. The periodic unfolding method in homogenization. *SIAM Journal on Mathematical Analysis*, 40(4):1585–1620, 2008.

[24] P. L. Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208, 1993.

[25] S. Conti, F. Hoffmann, and M. Ortiz. Model-free data-driven inference. *arXiv preprint arXiv:2106.02728*, 2021.

[26] S. Conti, S. Müller, and M. Ortiz. Data-driven problems in elasticity. *Archive for Rational Mechanics and Analysis*, 229(1):79–123, 2018.

[27] S. Conti, S. Müller, and M. Ortiz. Data-driven finite elasticity. *Archive for Rational Mechanics and Analysis*, 237(1):1–33, 2020.

[28] R. Courant. *Variational methods for the solution of problems of equilibrium and vibrations*. Bull. Amer. Math. Soc., 1943.

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

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

[31] M. Dalémat, M. Coret, A. Leygue, and E. Verron. Robustness of the data-driven identification algorithm with incomplete input data. *HAL Id : hal-03028848, version 3*, 2021.

[32] A. Der Kiureghian and J.-B. Ke. The stochastic finite element method in structural reliability. In *Stochastic structural mechanics*, pages 84–109. Springer, 1987.

[33] A. Der Kiureghian and J.-B. Ke. The stochastic finite element method in structural reliability. *Probabilistic engineering mechanics*, 3(2):83–91, 1988.

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

[35] 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*, 382:113855, 2021.

[36] 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*, 373:113499, 2021.

[37] H. C. Elman, O. G. Ernst, D. P. O'Leary, and M. Stewart. Efficient iterative algorithms for the stochastic finite element method with application to acoustic scattering. *Computer Methods in Applied Mechanics and Engineering*, 194(9-11):1037–1055, 2005.

[38] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[39] F. Feyel and J.-L. Chaboche. Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. *Computer methods in applied mechanics and engineering*, 183(3-4):309–330, 2000.

[40] D. Finol, Y. Lu, V. Mahadevan, and A. Srivastava. Deep convolutional neural networks for eigenvalue problems in mechanics. *International Journal for Numerical Methods in Engineering*, 118(5):258–275, 2019.

[41] W. Gawehn and S. Funk. *Finite Elemente Methode – FEM Grundlagen zur Statik und Dynamik*. Hochschule Osnabrück, 2017, 2017.

[42] M. Gioffre, V. Gusella, and M. Grigoriu. Simulation of non-gaussian field applied to wind pressure fluctuations. *Probabilistic Engineering Mechanics*, 15(4):339–345, 2000.

[43] H. B. M. GmbH. Temperaturkompensation bei dehnungsmessstreifen-messungen. Website. available online at https://www.hbm.com/de/6725/temperaturkompensation-bei-dehnungsmessstreifen-messungen/.

[44] K. I. GmbH. Datenblatt mehrkomponenten-kraftmesselement-bausatz. Website. available online at https://www.kistler.com/?type=669&fid=44722&model=document.

[45] H. Gödner. Lehrbuch Höhere Festigkeitslehre. VEB Fachbuchverlag Leipzig, 1984.

[46] D. Gonzalez, F. Chinesta, and E. Cueto. Learning corrections for hyperelastic models from data. *Frontiers in Materials*, 6:14, 2019.

[47] A. Haldar and S. Mahadevan. *Reliability assessment using stochastic finite element analysis*. John Wiley & Sons, 2000.

[48] Y. Heider, K. Wang, and W. C. Sun. So (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials. *Computer Methods in Applied Mechanics and Engineering*, 363:112875, 2020.

[49] A. J. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

[50] D. Z. Huang, K. Xu, C. Farhat, and E. Darve. Learning constitutive relations from indirect observations using deep neural networks. *Journal of Computational Physics*, 416:109491, 2020.

[51] T. J. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[52] 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*, 25(1):47–57, 2018.

[53] R. Ibáñez, E. Abisset-Chavanne, D. González, J.-L. Duval, E. Cueto, and F. Chinesta. Hybrid constitutive modeling: data-driven learning of corrections to plasticity models. *International Journal of Material Forming*, pages 1–9, 2018.

[54] 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*, 60(5):813–826, 2017.

[55] S. Jha, R. Brockman, R. Hoffman, V. Sinha, A. Pilchak, W. Porter, D. Buchanan, J. Larsen, and R. John. A data analytics approach to discovering unique microstructural configurations susceptible to fatigue. *JOM*, pages 1–7, 2018.

[56] O. L. Kafka, C. Yu, M. Shakoor, Z. Liu, G. J. Wagner, and W. K. Liu. Data-driven mechanistic modeling of influence of microstructure on high-cycle fatigue life of nickel titanium. *JOM*, pages 1–5, 2018.

[57] W. Kaiser. *Kunststoffchemie für Ingenieure: von der Synthese bis zur Anwendung*. Carl Hanser Verlag GmbH Co KG, 2015.

[58] S. Kanaun and O. Tkachenko. Representative volume element and effective elastic properties of open cell foam materials with random microstructures. *Journal of Mechanics of Materials and Structures*, 2(8):1607–1628, 2007.

[59] Y. Kanno. Data-driven computing in elasticity via kernel regression. *Theoretical and Applied Mechanics Letters*, 8(6):361–365, 2018.

[60] Y. Kanno. Simple heuristic for data-driven computational elasticity with material data involving noise and outliers: a local robust regression approach. *Japan Journal of Industrial and Applied Mathematics*, 35(3):1085–1101, 2018.

[61] Y. Kanno. Alternating minimization for data-driven computational elasticity from experimental data: kernel method for learning constitutive manifold. *Theoretical and Applied Mechanics Letters*, page 100289, 2021.

[62] Y. Kanno. A kernel method for learning constitutive relation in data-driven computational elasticity. *Japan Journal of Industrial and Applied Mathematics*, 38(1):39–77, 2021.

[63] D. G. . C. KG. Datenblatt zamak-5. Website. available online at `http://www.drutec-druckguss.de/pdf_dateien/datenblatt.pdf`.

[64] M. Khosravani and K. Weinberg. Experimental investigations of the environmental effects on stability and integrity of composite sandwich t-joints. *Materialwissenschaft und Werkstofftechnik*, 48(8):753–759, 2017.

[65] M. R. Khosravani and K. Weinberg. Characterization of sandwich composite t-joints under different ageing conditions. *Composite Structures*, 197:80–88, 2018.

[66] T. Kirchdoerfer. *Data Driven Computing*. PhD thesis, California Institute of Technology, 2017.

[67] T. Kirchdoerfer and M. Ortiz. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 304:81–101, 2016.

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

[69] T. Kirchdoerfer and M. Ortiz. Data-driven computing in dynamics. *International Journal for Numerical Methods in Engineering*, 113(11):1697–1710, 2018.

[70] T. Korzeniowski and K. Weinberg. A markov chain approach to damage evolution in die-cast zamak. *Technische Mechanik-European Journal of Engineering Mechanics*, 38(2):135–147, 2018.

[71] T. F. Korzeniowski, T. Reppel, and K. Weinberg. A juxtaposition of data driven and stochastic finite element analyses for problems with noisy material data. *PAMM*, 19(1):e201900197, 2019.

[72] 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*, 350:554–570, 2019.

[73] T. F. Korzeniowski and K. Weinberg. Data-driven diffusion with uncertainties. *PAMM*, 20(1):e202000325, 2021.

[74] T. F. Korzeniowski and K. Weinberg. Data-driven finite element computation of open-cell foam structures. *Computer Methods in Applied Mechanics and Engineering*, subm., 2021.

[75] T. F. Korzeniowski and K. Weinberg. A multi-level method for data-driven finite element computations. *Computer Methods in Applied Mechanics and Engineering*, 379:113740, 2021.

[76] L. Ladicky, S. Jeong, B. Solenthaler, M. Pollefeys, M. Gross, et al. Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics (TOG)*, 34(6):199, 2015.

[77] K. H. Lee. *First course on fuzzy theory and applications*, volume 27. Springer Science & Business Media, 2004.

[78] A. Leygue, M. Coret, J. Réthoré, L. Stainier, and E. Verron. Data driven constitutive identification. *preprint hal-01452494v2*, 2017.

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

[80] 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*, 64(2):501–509, 2019.

[81] A. Liebscher. Laguerre approximation of random foams. *Philosophical Magazine*, 95(25):2777–2792, 2015.

[82] C. Loop. Smooth spline surfaces over irregular meshes. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 303–310, 1994.

[83] X. Lu, D. G. Giovanis, J. Yvonnet, V. Papadopoulos, F. Detrez, and J. Bai. A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites. *Computational Mechanics*, 64(2):307–321, 2019.

[84] M. Marvi-Mashhadi, C. Lopes, and J. LLorca. Modelling of the mechanical behavior of polyurethane foams by means of micromechanical characterization and computational homogenization. *International Journal of Solids and Structures*, 146:154–166, 2018.

[85] H. G. Matthies, C. E. Brenner, C. G. Bucher, and C. G. Soares. Uncertainties in probabilistic numerical analysis of structures and solids-stochastic finite elements. *Structural Safety*, 19(3):283 – 336, 1997.

[86] C. Miehe, J. Schröder, and J. Schotte. Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials. *Computer methods in applied mechanics and engineering*, 171(3-4):387–418, 1999.

[87] F. Murtagh and K. Devlin. The development of data science: implications for education, employment, research, and the data revolution for sustainable development. *Big Data and Cognitive Computing*, 2(2):14, 2018.

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

[89] 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*, 365:112898, 2020.

[90] V. M. Nguyen-Thanh, X. Zhuang, and T. Rabczuk. A deep energy method for finite deformation hyperelasticity. *European Journal of Mechanics-A/Solids*, page 103874, 2019.

[91] W. G. Niklas Schietzold and M. Kaliske. Mehrkriterienoptimierung von Tragwerken aus Holz mit polymorph unscharfen Parametern . SPP1886 Annual Meeting, 2017.

[92] V. Papadopoulos and D. G. Giovanis. *Stochastic Finite Element Methods: An Introduction*. Springer, 2017.

[93] A. Platzer, A. Leygue, and L. Stainier. Assessment of data-driven computational mechanics in finite strain elasticity. In *Constitutive Models for Rubber XI*, pages 230–236. CRC Press, 2019.

[94] 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*, 379:113756, 2021.

[95] T. Reppel. *Flexible Materialien für die additive Fertigung: hyperelastische, viskoelastische Eigenschaften und der Einfluss unsicherer Materialparameter*. PhD thesis, University of Siegen, 2019.

[96] T. Reppel, T. Dally, and K. Weinberg. On the elastic modeling of highly extensible polyurea. *Technische Mechanik-European Journal of Engineering Mechanics*, 33(1):19–33, 2013.

[97] J. Réthoré, A. Leygue, M. Coret, L. Stainier, and E. Verron. Computational measurements of stress fields from digital images. *International Journal for Numerical Methods in Engineering*, 113(12):1810–1826, 2018.

[98] K. Sagiyama and K. Garikipati. Machine learning materials physics: Deep neural networks trained on elastic free energy data from martensitic microstructures predict homogenized stress fields with high accuracy. *arXiv preprint arXiv:1901.00524*, 2019.

[99] N. Scheerlinck, P. Verboven, J. D. Stigter, J. D. Baerdemaeker, J. F. V. Impe, and B. M. Nicolai. Stochastic finite-element analysis of coupled heat and mass transfer problems with random field parameters. *Numerical Heat Transfer, Part B: Fundamentals*, 37(3):309–330, 2000.

[100] K. Schulz, S. Kreis, H. Trittenbach, and K. Böhm. Data-driven crack assessment based on surface measurements. *Engineering Fracture Mechanics*, 218:106552, 2019.

[101] D. M. Sengeh, K. M. Moerman, A. Petron, and H. Herr. Multi-material 3-d viscoelastic model of a transtibial residuum from in-vivo indentation and mri data. *Journal of the mechanical behavior of biomedical materials*, 59:379–392, 2016.

[102] A. D. Spear, S. R. Kalidindi, B. Meredig, A. Kontsos, and J.-B. le Graverend. Data-driven materials investigations: The next frontier in understanding and predicting fatigue behavior. *JOM*, pages 1–4, 2018.

[103] L. Stainier, A. Leygue, and M. Ortiz. Model-free data-driven methods in mechanics: material data identification and solvers. *Computational Mechanics*, 64(2):381–393, 2019.

[104] G. Stefanou. The stochastic finite element method: past, present and future. *Computer Methods in Applied Mechanics and Engineering*, 198(9-12):1031–1051, 2009.

[105] B. Sudret and A. Der Kiureghian. *Stochastic finite element methods and reliability: a state-of-the-art report*. Department of Civil and Environmental Engineering, University of California Berkeley, 2000.

[106] S. Tang, Y. Li, H. Qiu, H. Yang, S. Saha, S. Mojumder, W. K. Liu, and X. Guo. Map123-ep: A mechanistic-based data-driven approach for numerical elastoplastic analysis. *Computer Methods in Applied Mechanics and Engineering*, 364:112955, 2020.

[107] S. Tang, H. Yang, H. Qiu, M. Fleming, W. K. Liu, and X. Guo. Map123-epf: A mechanistic-based data-driven approach for numerical elastoplastic modeling at finite strain. *Computer Methods in Applied Mechanics and Engineering*, 373:113484, 2021.

[108] S. Tang, G. Zhang, H. Yang, Y. Li, W. K. Liu, and X. Guo. Map123: A data-driven approach to use 1d data for 3d nonlinear elastic materials modeling. *Computer Methods in Applied Mechanics and Engineering*, 357:112587, 2019.

[109] S. Timoshenko and J. N. Goodier. Theory of Elasticity. McGraw-Hill Book Company, 1951.

[110] H. Tran, F. Charleux, M. Rachik, A. Ehrlacher, and M. Ho Ba Tho. In vivo characterization of the mechanical properties of human skin derived from mri and indentation techniques. *Computer methods in biomechanics and biomedical engineering*, 10(6):401–407, 2007.

[111] M. Wagner. Lineare und nichtlineare fem. *ISBN 978-3-658-17865-9. Springer Fachmedien Wiesbaden GmbH, 2017*, 2017.

[112] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.

[113] I. Wladawsky-Berger. The wall street journal; why do we need data science when we've had statistics for centuries? `https://blogs.wsj.com/cio/2014/05/02/why-do-we-need-data-science-when-weve-had-statistics-for-centuries/`. Accessed: 2019-12-11.

[114] C. F. J. Wu. Statistics = data science? `https://web.archive.org/web/20141014022300/http://www2.isye.gatech.edu/~jeffwu/presentations/datascience.pdf`. Accessed: 2019-12-11.

[115] O. H. Yeoh. Characterization of elastic properties of carbon-black-filled rubber vulcanizates. *Rubber chemistry and technology*, 63(5):792–805, 1990.

[116] L. A. Zadeh. Fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, pages 394–432. World Scientific, 1996.

[117] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Zhu. *The finite element method*, volume 3. McGraw-hill London, 1977.

[118] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[119] T. I. Zohdi, Zohdi, and Ditzinger. *A Finite Element Primer for Beginners*. Springer, 2018.

A Appendix

Equivalence of the principle of minimum potential energy and variational approach

To derive the weak form

$$\int_\Omega \delta\boldsymbol{\epsilon} : \boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{b}\cdot\delta\boldsymbol{u}\,\mathrm{d}\Omega - \int_{\Gamma_\sigma} \boldsymbol{t}\cdot\delta\boldsymbol{u}\,\mathrm{d}\Gamma$$

from the strong form

$$\mathrm{div}(\boldsymbol{\sigma}) + \boldsymbol{b} = 0 \tag{A.1}$$

we first multiply the balance of linear momentum with an arbitrary smooth function $\mathbf{v}$ and integrate over the whole domain

$$\int_\Omega (\mathrm{div}(\boldsymbol{\sigma}) + \boldsymbol{b})\cdot\boldsymbol{v}\,\mathrm{d}\Omega = 0 \tag{A.2}$$

The function $\boldsymbol{v}$ is often called test function and can be interpreted as a virtual displacement. Using the product rule of differentiation

$$\nabla\cdot(\boldsymbol{\sigma}\cdot\boldsymbol{v}) = (\nabla\cdot\boldsymbol{\sigma})\cdot\boldsymbol{v} + \nabla\boldsymbol{v} : \boldsymbol{\sigma}$$

for the first term we get

$$\begin{aligned}&\int_\Omega \nabla\cdot(\boldsymbol{\sigma}\cdot\boldsymbol{v}) - \nabla\boldsymbol{v} : \boldsymbol{\sigma}\,\mathrm{d}\Omega + \int_\Omega \boldsymbol{b}\cdot\boldsymbol{v}\,\mathrm{d}\Omega = 0\\ \Leftrightarrow &\int_\Omega \nabla\boldsymbol{v} : \boldsymbol{\sigma} - \nabla\cdot(\boldsymbol{\sigma}\cdot\boldsymbol{v})\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{b}\cdot\boldsymbol{v}\,\mathrm{d}\Omega = 0\end{aligned}$$

Using the divergence theorem by Gauss

$$\int_\Omega \mathrm{div}\,\boldsymbol{a}\,\mathrm{d}\Omega = \int_\Gamma \boldsymbol{a}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma$$

we obtain

$$\int_\Omega \nabla\boldsymbol{v} : \boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Gamma \boldsymbol{\sigma}\cdot\boldsymbol{n}\cdot\boldsymbol{v}\,\mathrm{d}\Gamma - \int_\Omega \boldsymbol{b}\cdot\boldsymbol{v}\,\mathrm{d}\Omega = 0.$$

As $\boldsymbol{v}$ can be seen as a virtual displacement it needs to vanish at the Dirichlet boundary such that the integral over the boundary reduces to the Neumann boundary where we can use the Neumann boundary condition $\mathbf{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$

$$\int_\Omega \nabla \boldsymbol{v} : \boldsymbol{\sigma} \, \mathrm{d}\Omega - \int_\Omega \boldsymbol{b} \cdot \boldsymbol{v} \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \boldsymbol{v} \, \mathrm{d}\Gamma = 0$$

which is the weak formulation of the problem as $\boldsymbol{v}$ equals $\delta \boldsymbol{u}$ of the Lagrangian notation in (2.10).

Function spaces

Another question we can ask is to which function spaces do the solution and the other variables belong. The weak formulation is called „weak“ as it weakens the differential conditions onto the solution. While in the strong form the solution $\mathbf{u}$ needed to be two times differentiable it only need to have derivatives of first order in the weak form. Furthermore it need to be ensured that the integrals in the weak form still exist

$$\int_\Omega \nabla \boldsymbol{v} : \boldsymbol{\sigma} \, \mathrm{d}\Omega < \infty \,, \quad \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}\Omega < \infty \,, \quad \int_{\Gamma_1} \mathbf{t} \cdot \boldsymbol{v} \, \mathrm{d}\Gamma_1 < \infty \,.$$

As we have to do with products the functions have to be square integrable by Hölder's inequality. Following we define some of the generally used function spaces. The L^2-space is the space of all functions which are square-integrable

$$L^2(\Omega) := \{ g : \Omega \to \mathbb{R} \ : \int_\Omega |g|^2 < \infty \}$$

By a bold notation we mean the functions which takes values in multi-dimensional vector spaces

$$\mathbf{L}^2(\Omega) := \{ g : \Omega \to \mathbb{R}^d \ : \int_\Omega |g|^2 < \infty \}$$

while the dimension d equals three most of the times in our case. Generally, we get the $\mathbf{L}^p$-space by replacing both twos by p

$$\mathbf{L}^p(\Omega) := \{ g : \Omega \to \mathbb{R}^d \ : \int_\Omega |g|^p < \infty \}$$

If we also demand that all mixed partial derivatives of maximum order k are in $\mathbf{L}^p$ we arrive at the Sobolev-spaces

$$\mathbf{W}^{k,p}(\Omega) = \{ g \in \mathbf{L}^p(\Omega) \ : D^\alpha g \in \mathbf{L}^p(\Omega) \ \forall |\alpha| \leq k \}$$

For the case of $p = 2$ the Sobolev-space $\mathbf{W}^{k,2}$ is also a Hilbert-space which we name $\mathbf{H}^k(\Omega)$ then. For the case $k = 1$ the norm induced by the scalar product is

$$\sqrt{\langle g, g\rangle}_{\mathbf{H}^1(\Omega)} = ||g||_{\mathbf{H}^1(\Omega)} = \left(\int_\Omega |g|^2 + |\nabla g|^2 \,\mathrm{d}\Omega\right)^{1/2}$$

As mentioned before the displacement need to be differentiable and also the derivatives need to be square-integrable. That means that the Hilbert-space $\mathbf{H}^1(\Omega)$ is the space in where we are looking for a solution. Often also some subspaces are defined for functions which fulfil the essential boundary conditions

$$\mathbf{H}^1_{\Gamma_0}(\Omega) = \{g \in \mathbf{H}^1(\Omega) \,:\, g|_{\Gamma_0} = \bar{\boldsymbol{u}}\}$$

or vanish at the essential boundary

$$\mathbf{H}^1_0(\Omega) = \{g \in \mathbf{H}^1(\Omega) \,:\, g|_{\Gamma_0} = 0\}$$

The second space is the space which we select for the test functions as we are not allowed to apply a virtual displacement where essential boundary conditions are enforced but need the same differential and integral conditions as before. The weak form then reads as follows

Find $\boldsymbol{u} \in \mathbf{H}^1_{\Gamma_0}(\Omega)$ such that $\forall \boldsymbol{v} \in \mathbf{H}^1_0(\Omega)$

$$\int_\Omega \nabla \boldsymbol{v} : \boldsymbol{\sigma} \,\mathrm{d}\Omega = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \,\mathrm{d}\Omega + \int_{\Gamma_1} \mathbf{t} \cdot v \,\mathrm{d}\Gamma$$

For the loads we also need to assume to be square-integrable over their corresponding space $\boldsymbol{f} \in \mathbf{L}^2(\Omega)$ and $\boldsymbol{t} \in \mathbf{L}^2(\Gamma_1)$, however, loads that are not smooth can also be considered without many problems [119].

Minimal property

Strictly speaking there is no evidence until now that the solution of $\delta\Pi = 0$ minimizes the potential energy. For a minimum it is sufficient to show that

$$\delta^2\Pi > 0\,.$$

This is quickly shown in our case with help of the second variation. As the only the first term of Π inhabits a quadratic term

$$\Pi = \frac{1}{2}\int_\Omega \boldsymbol{\epsilon} : \mathbb{C} : \boldsymbol{\epsilon} - \int_\Omega \boldsymbol{b} \cdot \boldsymbol{u} \,\mathrm{d}\Omega - \int_{\Gamma_1} \boldsymbol{t} \cdot \mathbf{u} \,\mathrm{d}\Gamma$$

the others vanish and we obtain

$$\delta^2\Pi = \int_\Omega \delta\boldsymbol{\epsilon} : \mathbb{C} : \delta\epsilon \, \mathrm{d}\Omega$$

By the positive definiteness of $\mathbb{C}$ the integrand is positive and so is the integral. This is sufficient.

Data-driven problem formulation

The local continuous problem at a fixed location is given by

$$\min_{\mathrm{y}\in\mathcal{C}(\boldsymbol{x}_0)} \min_{\mathrm{z}\in\mathcal{D}(\boldsymbol{x}_0)} ||\mathbf{y}-\mathbf{z}||_{\mathbb{C}^\circ(\boldsymbol{x}_0)}$$

with the given norm

$$||(\boldsymbol{\epsilon},\boldsymbol{\sigma})||_{\mathbb{C}^\circ(\boldsymbol{x}_0)} = \Big(\boldsymbol{\epsilon}(\boldsymbol{x}_0) : \mathbb{C}^\circ(\boldsymbol{x}_0) : \boldsymbol{\epsilon}(\boldsymbol{x}_0) + \boldsymbol{\sigma}(\boldsymbol{x}_0) : (\mathbb{C}^\circ)^{-1}(\boldsymbol{x}_0) : \boldsymbol{\sigma}(\boldsymbol{x}_0)\Big)^{1/2}$$

The global continuous problem is described by

$$\min_{\mathrm{y}\in\mathcal{C}} \min_{\mathrm{z}\in\mathcal{D}} ||\mathbf{y}-\mathbf{z}||_{\mathbb{C}^\circ(\boldsymbol{x})}$$

with the given norm

$$||\mathbf{z}(\boldsymbol{x})|| = \left(\int_\Omega \boldsymbol{\epsilon}(\boldsymbol{x}) : \mathbb{C}^\circ(\boldsymbol{x}) : \boldsymbol{\epsilon}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) : (\mathbb{C}^\circ)^{-1}(\boldsymbol{x}) : \boldsymbol{\sigma}(\boldsymbol{x}) \, \mathrm{d}\Omega\right)^{1/2}$$

The local discretized problem is described by

$$\min_{\mathrm{y}\in\mathcal{C}_m} \min_{\mathrm{z}\in\mathcal{D}_m} ||\mathbf{y}-\mathbf{z}||_{\mathbb{C}_m^\circ}$$

with the given norm

$$||(\boldsymbol{\epsilon},\boldsymbol{\sigma})||_{\mathbb{C}_m^\circ} = \Big(\boldsymbol{\epsilon} : \mathbb{C}_m^\circ : \boldsymbol{\epsilon} + \boldsymbol{\sigma} : (\mathbb{C}_m^\circ)^{-1} : \boldsymbol{\sigma}\Big)^{1/2}$$

Stationary Gaussian fields

Gaussian fields are very attractive because they are defined by a few parameters. Namely, these are the expected value function and the autocovariance function. Furthermore, for Gaussian fields it holds that stationarity in a wide sense is equivalent to stationarity in a strict sense. Generally, stationarity in a wide sense does not imply stationarity in

a strict sense. The behavior of stationary processes at two different locations depends solely on the distance then. The stochastic process χ is said to be strictly stationary if

$$F_\chi(x_{t_1}, \ldots, x_{t_n}) = F_\chi(x_{t_1+\tau}, \ldots, x_{t_n+\tau}) \quad \forall \tau, t_1, \ldots, t_n \in \mathbb{R}, n \in \mathbb{N}$$

where F is the cumulative distribution function of the joint distribution at given times or locations.

System of equations and material data of Chapter 6

Systems of equations for the F,P formulation

The linear finite element system of Eq. (**??**) and (**??**) reads:

$$\begin{aligned}
\mathbf{K}_u^e \hat{\boldsymbol{u}}_e = \mathbf{f}_u^e \ : \quad & \mathbf{K}_u^e = \mu_0 \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{B}^e \, \mathrm{d}\Omega && \mathbf{K}_u = \bigcup_{\mathcal{E}} \mathbf{K}^e \\
& \mathbf{f}_u^e = \mu_0 \int_{\Omega^e} \mathbf{B}^e \boldsymbol{\epsilon}^* \, \mathrm{d}\Omega && \mathbf{f}_u = \bigcup_{\mathcal{E}} \mathbf{f}^e \\
\mathbf{K}_\lambda^e \hat{\boldsymbol{\lambda}}_e = \mathbf{f}_\lambda^e \ : \quad & \mathbf{K}_\lambda^e = \mu_0 \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{B}^e \, \mathrm{d}\Omega && \mathbf{K}_\lambda = \bigcup_{\mathcal{E}} \mathbf{K}^e \\
& \mathbf{f}_\lambda^e = \mathbf{f}^e - \int_{\Omega^e} \mathbf{B}^{eT} \boldsymbol{\sigma}^* \, \mathrm{d}\Omega && \mathbf{f}_\lambda = \bigcup_{\mathcal{E}} \mathbf{f}^e
\end{aligned}$$

Systems of equations for the C,S formulation

The non-linear finite element system of Eq. (6.20) reads as residual equations

$$\begin{aligned}
\mathbf{R}_u &= \int_{\Omega_e} \mathbf{B}^{eT} \left(2\mu_0 \mathbf{F}^{eT} \left(\mathbf{F}^{eT} \mathbf{F}^e - \mathbf{C}^* \right) - \mathbf{B}^{eT} \hat{\boldsymbol{\lambda}} \mathbf{S}^* - \mu_0 \mathbf{B}^{eT} \hat{\boldsymbol{\lambda}} \mathbf{B}^e \hat{\boldsymbol{\lambda}} \mathbf{F}^{eT} \right) \mathrm{d}\Omega = 0 \\
\mathbf{R}_\lambda &= \int_{\Omega_e} \mu_0 \mathbf{B}^{eT} \mathbf{F}^e \mathbf{B}^e \hat{\boldsymbol{\lambda}} \mathbf{F}^{eT} + \mathbf{B}^{eT} \mathbf{F}^e \mathbf{S}^* \, \mathrm{d}\Omega - \mathbf{f}^{\mathrm{ext}} = 0
\end{aligned}$$

A corresponding Newton-Raphson iteration step has the form

$$\begin{bmatrix} \hat{\boldsymbol{u}}_{j+1} \\ \hat{\boldsymbol{\lambda}}_{j+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{u}}_j \\ \hat{\boldsymbol{\lambda}}_j \end{bmatrix} + \begin{bmatrix} \Delta\hat{\boldsymbol{u}} \\ \Delta\hat{\boldsymbol{\lambda}} \end{bmatrix} \quad \text{mit} \quad \begin{bmatrix} \mathbf{R}_u(\hat{\boldsymbol{u}}_j, \hat{\boldsymbol{\lambda}}_j) \\ \mathbf{R}_\lambda(\hat{\boldsymbol{u}}_j, \hat{\boldsymbol{\lambda}}_j) \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{u\lambda} \\ \mathbf{K}_{\lambda u} & \mathbf{K}_{\lambda\lambda} \end{bmatrix} \begin{bmatrix} \Delta\hat{\boldsymbol{u}} \\ \Delta\hat{\boldsymbol{\lambda}} \end{bmatrix} = \mathbf{0}$$

where the $\mathbf{K}$ terms abbreviate the current tangent stiffness matrix. Their entries are calculated as

$$\begin{aligned}
\mathbf{K}_{uu} = \frac{\partial \mathbf{R}_u}{\partial \mathbf{u}} &= \int_{\Omega_e} \mathbf{B}^{eT} \Big(2\mu_0 \mathbf{B}^{eT} (\mathbf{F}^{eT}\mathbf{F}^e - \mathbf{C}^*) \\
&\quad + 4\mu_0 \mathbf{F}^{eT}\mathbf{B}^{eT}\mathbf{F}^e - \mu_0 \mathbf{B}^{eT}\hat{\boldsymbol{\lambda}}\mathbf{B}^e\hat{\boldsymbol{\lambda}}\mathbf{B}^{eT} \Big) \, \mathrm{d}\Omega \\
\mathbf{K}_{u\lambda} = \frac{\partial \mathbf{R}_u}{\partial \boldsymbol{\lambda}} &= \int_{\Omega_e} -\mathbf{B}^{eT}\mathbf{B}^{eT}\mathbf{S}^* - \mathbf{B}^{eT}\mu_0\mathbf{B}^{eT}\mathbf{B}^e\hat{\boldsymbol{\lambda}}\mathbf{F}^{eT} \\
&\quad - \mathbf{B}^{eT}\mu_0\mathbf{B}^{eT}\hat{\boldsymbol{\lambda}}\mathbf{B}^e\mathbf{F}^{eT} \, \mathrm{d}\Omega \\
\mathbf{K}_{\lambda u} = \frac{\partial \mathbf{R}_\lambda}{\partial \mathbf{u}} &= \int_{\Omega_e} \mu_0\mathbf{B}^{eT}\mathbf{B}^e\mathbf{B}^e\hat{\boldsymbol{\lambda}}\mathbf{F}^{eT} + \mu_0\mathbf{B}^{eT}\mathbf{F}^e\mathbf{B}^e\hat{\boldsymbol{\lambda}}\mathbf{B}^{eT} \\
&\quad + \mathbf{B}^{eT}\mathbf{B}^{eT}\mathbf{S}^* \, \mathrm{d}\Omega \\
\mathbf{K}_{\lambda\lambda} = \frac{\partial \mathbf{R}_\lambda}{\partial \boldsymbol{\lambda}} &= \int_{\Omega_e} \mu_0\mathbf{B}^{eT}\mathbf{F}^e\mathbf{B}^e\mathbf{F}^{eT} \, \mathrm{d}\Omega
\end{aligned}$$

In each time or load step and in each data iteration the iterative solution of the system is necessary.

Data generation of the 3-D example

For the three dimensional example in Section **??** the six deformations of the linear case C are described by six homogenized strain values. Remind that in the linear regime $\boldsymbol{C} \approx 2\boldsymbol{\epsilon} + 1$ and we can also gain data of other strain and stress equivalents in this regime. Therefore, we adopt small strain notation which leads to the six strain tensors in Voigt notation

$$\bar{\boldsymbol{\epsilon}}^{(1)} = \begin{pmatrix} \alpha \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \bar{\boldsymbol{\epsilon}}^{(2)} = \begin{pmatrix} 0 \\ \alpha \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \bar{\boldsymbol{\epsilon}}^{(3)} = \begin{pmatrix} 0 \\ 0 \\ \alpha \\ 0 \\ 0 \\ 0 \end{pmatrix}, \bar{\boldsymbol{\epsilon}}^{(4)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \alpha \\ 0 \\ 0 \end{pmatrix}, \bar{\boldsymbol{\epsilon}}^{(5)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \alpha \\ 0 \end{pmatrix}, \bar{\boldsymbol{\epsilon}}^{(6)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \alpha \end{pmatrix} . \quad \text{(A.3)}$$

with an arbitrary deformation defined by $\alpha \in \mathbb{R}$. These are applied to the RVE in individual simulations via the boundary conditions. The six computations with the

introduced RVE lead with $\alpha = 0.02$ to the six homogenized stresses $\bar{\sigma}^{(i)}$, $i = 1, \ldots, 6$:

$$\bar{\sigma}^{(1)} = \begin{pmatrix} 9.13 & 3.87 & 3.96 & -0.00 & 0.00 & -0.00 \end{pmatrix}^T \cdot 10^4 \text{ Pa}$$

$$\bar{\sigma}^{(2)} = \begin{pmatrix} 3.87 & 8.57 & 3.89 & 0.00 & 0.00 & -0.00 \end{pmatrix}^T \cdot 10^4 \text{ Pa}$$

$$\bar{\sigma}^{(3)} = \begin{pmatrix} 3.96 & 3.89 & 8.89 & -0.00 & 0.00 & -0.00 \end{pmatrix}^T \cdot 10^4 \text{ Pa}$$

$$\bar{\sigma}^{(4)} = \begin{pmatrix} -0.00 & 0.00 & 0.00 & 1.94 & -0.00 & 0.00 \end{pmatrix}^T \cdot 10^4 \text{ Pa}$$

$$\bar{\sigma}^{(5)} = \begin{pmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 1.96 & 0.00 \end{pmatrix}^T \cdot 10^4 \text{ Pa}$$

$$\bar{\sigma}^{(6)} = \begin{pmatrix} -0.00 & -0.00 & -0.00 & 0.00 & -0.00 & 1.99 \end{pmatrix}^T \cdot 10^4 \text{ Pa}$$