Sensitivity and Optimisation of Damage in Forming Processes – Academic and Industrial Challenges

Dissertation

Fabian Guhr

Sensitivity and Optimisation of Damage in Forming Processes – Academic and Industrial Challenges

von der Fakultät Architektur und Bauingenieurwesen
der Technischen Universität Dortmund
zur Verleihung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von
Fabian Guhr

Dortmund, Juli 2024

Kontakt

Fabian Guhr, M. Sc.
Lehrstuhl Baumechanik
Fakultät Architektur und Bauingenieurwesen
Technische Universität Dortmund
August-Schmidt-Str. 8, D-44227 Dortmund
Email: fabian.guhr@tu-dortmund.de

Prüfungskommission

Erster Referent (first referee):	Prof. Dr.-Ing. Franz-Joseph Barthold
Zweiter Referent (second referee):	Prof. Dr.-Ing. Andreas Menzel
Vorsitz (committee chair):	Prof. Dr.-Ing. Ingo Münch

Tag der Einreichung (thesis submission):	15.03.2024
Tag der mündlichen Prüfung (thesis defense):	18.06.2024

"*The most important step a man can take. It's not the first one, is it? It's the* next *one. Always the next step.*"

– from Oathbringer by Brandon Sanderson

Abstract

Sensitivity and Optimisation of Damage in Forming Processes – Academic and Industrial Challenges

Manufacturing of engineering components in today's industry requires a great amount of knowledge in order to produce components with the desired capabilities. This knowledge encompasses the material behaviour, the manufacturing process which creates the part, as well as the loading which the product is subjected to during its intended use. Combination of those factors allows an engineer to design a product which satisfies the requirements for the task at hand. While elastic and elasto-plastic effects are well understood nowadays, damage mechanics take a more leading role in research. Damage can be understood as an accumulation of defects on the microscale, which leads to a degradation of the macroscopic behaviour of the material. Including damage modelling into the engineering process, e.g. by using damage material models for the simulations, it is possible to predict damage accumulation due to the manufacturing process. This damage can subsequently be reduced by adjusting process parameters, in order to generate parts which not only show overall lesser damage accumulation, but also further resistance against damage when in use. While typically such design processes are done manually, the utilisation of numerical optimisation allows automatic generation of damage resistant designs. Here, the use of numerical optimisation enables the automatic generation of damage-tolerant processes and components.

This thesis deals with numerical damage optimisation and consists of two overarching parts. The first part covers the academic challenges of an optimisation problem. A nonlocal ductile damage model is enhanced with sensitivity information to allow for shape optimisation. Analytical derivation of the gradients enables the application of gradient-based optimisation strategies which greatly reduce the computation time. While this approach is computational efficient and is elegant in deriving the sensitivity information, its scope of application is limited to academic problems. Consequently, the second part of this thesis deals with the industrial challenges. For this, an additional optimisation framework approach around a commercial finite element software is derived. This framework allows damage optimisation of forming processes, such that parameter combinations can be found which produce parts with less damage accumulation.

Key words: Nonlocal damage, finite element method, variational sensitivity analysis, structural optimisation, shape optimisation, forming processes, process optimisation, Abaqus.

Kurzfassung

Sensitivität und Optimierung der Schädigung in Umformprozessen – Akademische und industrielle Herausforderungen

Die Herstellung technischer Komponenten in der heutigen Industrie erfordert ein großes Maß an Wissen, um Teile mit den gewünschten Eigenschaften zu produzieren. Dieses benötige Wissen umfasst das Materialverhalten, den Herstellungsprozess, sowie die Belastung, welche das Bauteil im Einsatz erfährt. Eine Kombination dieser Faktoren ermöglicht es dem Ingenieur, ein Produkt zu entwerfen, welches den Anforderungen der jeweiligen Anwendung gerecht wird. Elastische und plastische Materialeigenschaften sind heutzutage reichlich erforscht. Schädigungsmechanik stellt jedoch weiterhin eine anspruchsvolle Herausforderung dar und nimmt eine wichtige Rolle in der aktuellen Forschung ein. Schädigung wird als eine Ansammlung von Defekten auf der Mikroskala verstanden, welche zu einer Verschlechterung des makroskopischen Materialverhaltens führt. Die Berücksichtigung von Schädigungsmodellierung im Konstruktionsprozess ermöglicht die Vorhersage des Schädigungszustandes, welcher durch den Fertigungsprozess entsteht. Die Schädigung kann durch Anpassung von Prozessparametern reduziert werden, um Teile zu erzeugen, die eine geringere Schädigungsanreicherung aufweisen und in der Anwendung schädigungstoleranter sind. Hier ermöglicht der Einsatz numerischer Optimierung die automatische Generierung schädigungstoleranter Prozesse und Bauteile.

Diese Arbeit befasst sich mit der numerischen Schädigungsoptimierung und lässt sich in zwei Bereiche einteilen. Der erste Bereich befasst sich mit den akademischen Herausforderungen eines solchen Optimierungsproblems. Ein nicht-lokales duktiles Schädigungsmodell wird mit Sensitivitätsinformationen angereichert, um Formoptimierung zu ermöglichen. Die analytische Herleitung der Gradienten erlaubt die Anwendung gradientenbasierter Optimierungsstrategien, wodurch die Rechenzeit erheblich reduziert wird. Dieser Ansatz ist zwar rechnerisch effizient und elegant für die Herleitung der Sensitivitätsinformationen, der Anwendungsbereich ist jedoch auf akademische Probleme beschränkt. Daher befasst sich der zweite Bereich dieser Arbeit mit den industriellen Herausforderungen. Dafür wird eine zusätzliche Optimierungsumgebung um eine kommerzielle Finite Elemente Software herum entwickelt. Diese ermöglicht eine Schädigungsoptimierung von Umformprozessen, sodass Parameterkombinationen gefunden werden können, die Bauteile mit geringerer Schädigungsanreicherung erzeugen.

Schlagwörter: Nichtlokale Schädigung, Finite Elemente Methode, Variationelle Sensitivitätsanalyse, Struktur- und Formoptimierung, Umformprozesse, Prozessoptimierung, Abaqus.

Vorwort

Diese Dissertation entstand während meiner Arbeit als wissenschaftlicher Mitarbeiter am Lehrstuhl Baumechanik des Instituts für Baumechanik, Statik und Dynamik der Fakultät Architektur und Bauingenieurwesen an der Technischen Universität Dortmund.

Beginnend möchte ich mich bei meinem Betreuer und Doktorvater Professor Franz-Joseph Barthold bedanken die Möglichkeit bekommen zu haben diese Arbeit zu schreiben. Darüber hinaus danke ich ihm insbesondere auch für die stetige Förderung, den Rückhalt und die Unterstützung bei komplexeren Problemen während meiner Promotionsphase. Das angenehme Arbeitsumfeld ermöglichte es mir die entstandenen Herausforderungen mit eigenen Ideen anzugehen.

Weiterhin möchte ich meinem Zweitprüfer Professor Andreas Menzel danken - sowohl für die exzellente Ausbildung die ich während meines Studiums im Maschinenbau erhalten habe, als auch für die Weiterempfehlung an Professor Barthold im Jahre 2016, womit mir die Promotion überhaupt erst ermöglicht wurde.

Ein großes Dankeschön geht sowohl an meine aktuellen als auch an meinen ehemaligen Arbeitskollegen am Institut. Spezieller Dank gilt meinem Büropartner Jan Liedmann für seine Hilfsbereitschaft bei mathematischen Fragen, sowie auch etwaigen Problemen mit Linux- oder Latex-Angelegenheiten. Ich möchte auch unserer Sekretärin Mirjana Vukovic für die administrative Arbeit und Markus Behlau für die Unterstützung im Bereich IT danken.

Über unser Institut hinaus möchte ich den aktuellen und ehemaligen Angestellten am Institut für Mechanik der Fakultät Maschinenbau in Dortmund danken. Meine akademische Laufbahn begann dort als studentische Hilfskraft und dabei konnte ich bereits Kontakte knüpfen, die mir später bei Konferenzen oder anderen Gelegenheiten erlaubten meinen wissenschaftlichen Horizont über mein Themengebiet hinaus zu erweitern.

Abschließend möchte ich mich noch besonders bei meinen Freunden und meiner Familie bedanken. Speziell bei meiner Schwester Jorina, die mir immer zusprach, wenn ich während meiner Promotion mit mir selbst haderte, aber insbesondere meinen Eltern Susanne und Joachim, die mich während meiner gesamten akademischen Laufbahn unterstützt haben und auf deren Rückhalt ich immer vertrauen konnte. Vielen Dank.

Dortmund, Juli 2024 Fabian Guhr

External work

Certain experimental investigations, as well as supply of modelling data were provided by other researchers in the course of this research. For sake of transparency, their contribution is listed in the following and additionally emphasised within the respective chapters. The assistance they provided is gratefully acknowledged.

Contributions:
Testing of 3d-printed specimen for damage optimised structures. Conducted by Leon Sprave, PhD student at the Institute of Mechanics (IOM), TU Dortmund University. (Appendix C.1.3)

Modelling of rod extrusion processes in Abaqus. Conducting of experiments and subsequent analysis of micro structure. Conducted by Robin Gitschel, PhD student at the Institute of Forming Technology and Lightweight Components (IUL), TU Dortmund University. (Chapter 10, Appendix D.1)

Modelling of elastomer bending in Abaqus. Provided by Rickmer Meya, PhD student at the Institute of Forming Technology and Lightweight Components (IUL), TU Dortmund University. (Chapter 10)

Modelling of stretch indenting in Abaqus. Provided by Martina Müller, PhD student at the Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University. (Chapter 10)

Contents

Chapter 1

Introduction

This chapter introduces the reader to the broad thematic background of the underlying research and motivates the topic of interest. The concept of damage in metals is presented and a distinction between different damage mechanisms is made. The field of structural optimisation is described and process optimisation is briefly introduced. The overarching goal of this thesis is further specified for the academic and industrial viewpoints, respectively. Finally, the structure of this is presented.

1.1 Motivation

Over the recent decades, the demands regarding metals in industrial applications has increased steadily. This led to specific microstructures or textures of the material being engineered and improved for their respective scope of application. The utilisation in their fields of manufacturing in return lead to higher demands on the manufactured parts, such as formability, lightweight structures, manufacturing cost and safety margins, to name a few. Together with ever growing concern regarding the environmental impact and therefore material and energy efficient production, the requirements in modern industry are ever increasing. This requires engineers to acquire a deep understanding of the underlying mechanics regarding material mechanics, as well as the considered process.

One of the key mechanisms which govern the efficient utilisation of metals is damage. If damage evolution is too high during the manufacturing process, macroscopic fracture may occur and failure of the part is highly likely. This macroscopic fracture is often seen as the critical factor in a finished product. In turn, microscopic damage accumulation is often not considered. However, unaccounted induced damage during manufacturing can negatively influence the behaviour and the lifetime of a given part. Hence, it is necessary to accurately research damage behaviour and analyse processes regarding the accumulation of damage during manufacturing. Taking damage effects into consideration during experiments for example, allows analysis of the influence certain process parameters have on the microscopic damage evolution during manufacturing. However, only utilising experimental data is quite cost inefficient.

Nowadays, simulation of physical problems is part of most industrial environments, due to ever increasing computational power. Consideration of damage in a given material model allows its computation in numerical simulations and therefore prediction of the degradation of material behaviour due to applied loads. Obviously, this is more cost efficient than relying solely on experimental data. However, modelling of damage behaviour is not straightforward, as it incorporates additional numerical challenges which require their own methods to solve.

With a predictive material model and a corresponding simulation environment at hand, numerical optimisation can be applied to further enhance a given problem. The underlying results from a process, in either the experimental setup or in simulations, directly depend on the choice of many process parameters and other design variables. By applying mathematical optimisation, optimal parameter sets can be generated which reduce the damage accumulation within the researched process. This motivates the work presented in this thesis. Combination of mathematical optimisation with ductile damage material mechanics enables enhancement of structures and forming processes to yield damage tolerant geometries and processes, respectively. The considered materials are DP800, used in sheet metal forming, and 16MnCrS5, used in bulk metal forming. Therefore, only ductile damage is the considered mechanism in this work.

1.1.1 Basics of damage mechanics

One of the most crucial material properties of a part under consideration is its damage state. Knowing how much damage is present in a product allows better utilisation of its properties as the limits regarding its load bearing capabilities are better predicted. Damage as a mechanism has been known in metal manufacturing for about a century. First investigations were undertaken by [150] in 1926, where tensile tests for aluminium samples were investigated. Following this early research, later on in 1930 [185] chevron-crack in wire drawing was discovered and in [42] the influence of hydrostatic pressure on ductile fracture was highlighted in 1945. First ground breaking innovations in continuum damage modelling came with the work done by [124] in 1958. In general, a distinction between three different cases of damage mechanics can be made, i.e. between *fatigue*, *brittle damage* and *ductile damage*.

Fatigue describes the loss of load bearing capacity due to damage accumulation under cycling loads [126]. Notably, this loss occurs at a prescribed load that has a lower magnitude than is observed for failure in non-cycling experiments. Depending on the number of cycles, it can be further differentiated between low-cycle fatigue and high-cycle fatigue. In general, fatigue is not explicitly applicable to forming processes for example, as its effects are only relevant for the later use case of manufactured components.

Brittle damage is a damage mechanism, which occurs without a coupled plastic behaviour. It is therefore generally observed in brittle and porous structures, which show no plastic deformations, such as ceramics, concrete or rock [65], or materials at extremely low temperatures. Generally, brittle damage can be identified by analysing a crack surface. Since brittle damage occurs without plastic effects, the surface of brittle fracture is relatively smooth in perpendicular orientation to the crack direction.

However, both of the aforementioned types of damage are not considered in this thesis. Since metals and forming processes are the choice of material in this work, the coupling of plasticity with damage effects have to be accounted for. Therefore, the main principle damage effect under consideration in this thesis is ductile damage.

Ductile damage describes the combined evolution of plasticity and damage. It is understood as the nucleation, growth and coalescence of pores or voids on the microscale, see Fig. 1.1. These generally initiate at micro defects within a given material, such as inclusions or grain boundaries [195, 219]. Since these voids grow due to plastic deformation [84], analysis and detection of damage initiation processes within the material are hard to distinguish. Generally, ductile damage within the microscale of the material results due to brittle fracture at hard phases in the material, decohesion of interfaces, or micro-cracks in martensite [66, 137, 242]. Comparing fracture surfaces, in contrast to brittle fracture, fracture resulting from ductile damage can be identified by a dimpled fracture surface.

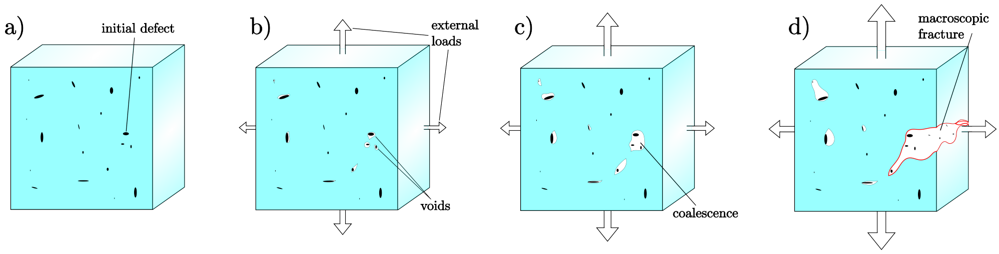


Figure 1.1: Sketch of ductile damage evolution under increasing load (based on [186]): a) The initial, inhomogeneous material, b) void nucleation, c) void growth and coalescence, and d) macroscopic fracture and failure.

Depending on the field of research, different definitions for damage can be found. In [114] a definition for damage is proposed that attempts to take into account the most important properties in the fields of process technologies, characterisation and material modelling. In this thesis, based on the work within TRR188 [224], damage is understood as an effect on the microscale, which negatively influences the macroscopic mechanical properties of the material under consideration. The difference between an undamaged and damaged material can be observed by their elastic properties under the same load. Application of new methodology enables decoupling and therefore detailed research of these mechanics. By means of in-situ methods for example, the underlying mechanisms can be analysed on the microscale directly [115].

1.1.2 Basics of applied optimisation

Utilisation of mathematical optimisation in numerics, specifically structural problems, is an established field in structural mechanics. Structural optimisation (SO) captures many different problem types, see e.g. [12, 27, 199] for a general overview. In this field of research, the aim is to reduce an objective function by altering the underlying geometry of the structural problem. Since the structural response of a system inherently depends on the initial geometry, this approach enables automatic generation of optimal geometric designs. Depending on the type of design variable, and thereby the possibility to change the geometry, the field of research can be further specified.

The simplest type of SO to consider, are so-called sizing optimisations. Therein, specific geometric dimensions of a model are changed, such as thicknesses or cross-sections, cf. [4, 87]. Due to its simplicity, implementation into industrial environments is straightforward.

One of the most important research topics within SO is the field of topology optimisation. Therein, within an initial continuous body, holes are added and removed throughout the optimisation, to generate a new structure. The well-known SIMP (Solid Isotropic Material with Penalisation) defines elemental densities as the design variables of the optimisation. By constraining their value between full material (density 1) and no material (density 0), holes can be placed into a predefined geometry. However, the topology of the structures considered in this work is continuous and therefore topology optimisation is not a topic of research here. For an overview, cf. [3, 28, 203, 204].

In contrast, shape optimisation aims to solve the proposed optimisation problem by changing the general shape of a structure. That is, no holes can be created nor removed, but instead the surfaces are adjusted to generate a new shape. Basics can be found in e.g. [17, 27, 35, 52, 125, 208]. The geometry in such cases is often described utilising Computer Aided Geometric Design (CAGD) [74] to generate smooth surfaces.

Besides SO, other fields of optimisation are also established in research. By applying optimisation schemes to processes, these can be optimised as well, and are denoted as *process optimisation* techniques. Due to the high amount of different forming processes, the possibilities regarding optimisation problems is extensive. For a small overview see e.g. [8, 38].

The underlying material description of the structural analysis is of high importance for the conducted optimisation. As only the simulated results can be optimised, higher predictive capabilities of the model allow for more accurate representation of the actual physics during optimisation. This is even more important for the field of parameter identification (PI). This field aims to correlate data from experiments with simulation results, such that the simulated data most accurately models the desired material behaviour. The design variables in these types of optimisations are generally the constitutive material parameters of the underlying material model. For more on this field of research, cf. [108, 134, 153, 190, 192]. Nonetheless, more complex, nonlinear problems are applied within existing research, such as elasto-plasticity [127, 145, 146, 233], damage [14, 99, 103, 175], multi-scale [129, 130], thermal effects [202], viscous effects [241] or finite elements of higher order [19, 236].

1.2 Goals of this work

The overarching objective of this work is to

Enable the minimisation and control of damage by applying the concept of mathematical optimisation.

The work presented in this thesis is based on research conducted within TRR188 “Damage Controlled Forming Processes” [224]. Since the main objective of this collaborative research centre is the control and minimisation of damage within forming processes, their consideration is also of vital importance for this work. Application of mathematical optimisation for damage problems in this work has to be considered from two viewpoints. First, the academic viewpoint and their challenges and second, the consideration of forming processes within mathematical optimisation from the industrial viewpoint. These challenges, and their respective methodologies, are presented in the two overarching parts of this work. That is, Part I: Academic challenges and Part II: Industrial challenges illustrate their own approaches to apply mathematical optimisation to damage problems. Both challenges and their methodology are graphically depicted in Fig. 1.2.

Mathematical optimisation is utilised to optimise damage in the arising problems of both viewpoints. Therefore, a generalised formulation for the mathematical optimisation problem is required which describes the problems in both of these viewpoints. This prescribed problem consists of an objective and possible constraints, and depends on a set of design variables which govern the underlying model. Solution strategies are required to solve the mathematical optimisation within the numerical setting. Gradient-based optimisation techniques are very efficient to solve such optimisation problems. However, under certain circumstances, gradient-free methods might be more suitable, hence both strategies have to be considered.

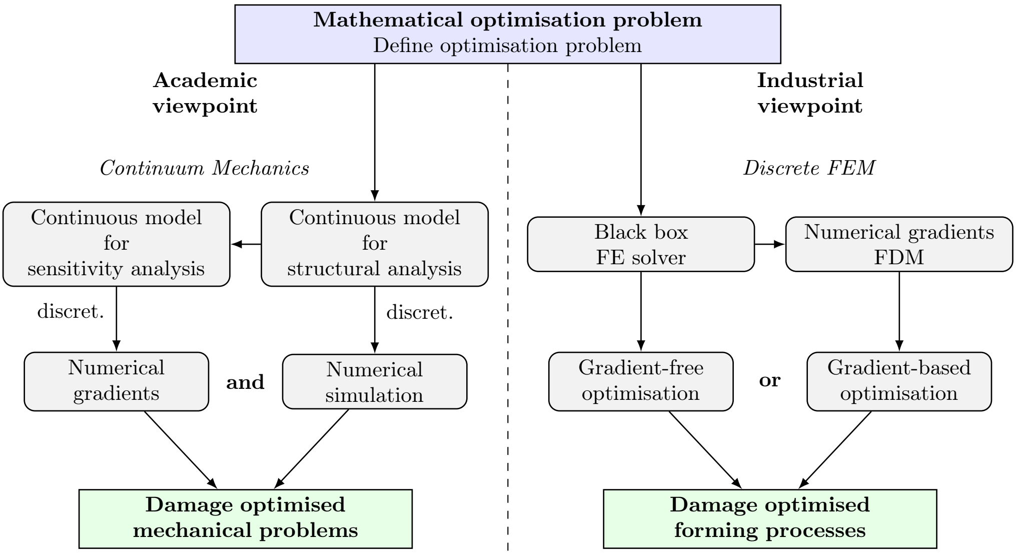

Mathematical optimisation problem
Define optimisation problem
Academic viewpoint
Industrial viewpoint
Continuum Mechanics
Discrete FEM
Continuous model for sensitivity analysis
Continuous model for structural analysis
Black box FE solver
Numerical gradients FDM
discret.
discret.
Numerical gradients
and
Numerical simulation
Gradient-free optimisation
or
Gradient-based optimisation
Damage optimised mechanical problems
Damage optimised forming processes

Figure 1.2: Overview of the academic and industrial viewpoints.

The aim within **Part I: Academic challenges** is to enhance a continuum mechanical problem by applying sensitivity analysis. This enhancement, coupled with an efficient implementation, allows utilisation of gradient-based solution strategies to the optimisation problem. A macroscopic ductile damage material model has to be implemented to predict the damage behaviour within numerical simulations. Enhancement of this model in its continuous description, i.e. prior to discretisation, enables efficient derivation of sensitivity information. The *academic challenges* herein are:

Enhancement of the continuum description for the ductile damage model
by means of analytical sensitivity analysis.

Utilising the knowledge from the continuous formulation of the material model, analytical gradients can be deduced on the continuous level and circumvent application of numerically derived gradients. This is favourable compared to computation of numerical gradients by means of the finite difference method (FDM). Due to the underlying multi-field approach and the dissipative, history dependent nature of the ductile damage, these effects are also important for the applied sensitivity analysis. After derivation by hand, these continuous gradients are discretised for the numerical implementation. They find application within mathematical optimisation to enable the control and minimisation of damage utilising numerical methods.

Application of the methodology from the academic viewpoint in Part I, i.e. analytical derivation of sensitivity information, to the industrial viewpoint in Part II is a very challenging task. The research within TRR188 is focused on forming processes. Hence, the methodology of deriving analytical gradient information for these processes, which inherently require contact formulations to be accurately modelled, is very time intensive. Therefore, to tackle these industrial forming processes within mathematical optimisation, another approach is preferred. The objective of in **Part II: Industrial challenges** and this second viewpoint, is to generate an optimisation framework which can handle the arising optimisation problems explicitly considering forming processes. The main task for the *industrial challenges* therefore is:

Formulation of an optimisation framework which enables the minimisation and control of damage within the forming processes of TRR188.

Therefore, this methodology aims to be directly applicable to any process, without prior deep understanding of the underlying continuum material model. Implicitly, contact mechanics have to be considered within the optimisation. Furthermore, many different underlying material models are utilised in the different process, e.g. some processes have to consider thermo-mechanical material behaviour, while other consider elasto-plastic behaviour. For this viewpoint, a black-box approach for the material description and simulation has to be utilised. This means, the underlying continuous description of the continuum material model is unknown. Hence, the gradients cannot be derived analytically and have to be computed by means of the FDM. Due to the discontinuous contact mechanics, these gradients might be insufficient for gradient-based solution strategies, requiring the additional consideration of gradient-free methods. The considered forming process within the optimisations are widely utilised in the industry.

1.3 Outline

The work presented in this thesis deals with applying numerical optimisation to two different fields of research, i.e. an academic viewpoint and a more industrial viewpoint. The thesis is therefore structured in two parts which consider each respective field of application. Since both fields utilise numerical optimisation, the concept is briefly introduced to present the notation used within this work. Solution strategies to solve constrained and unconstrained optimisation problems are presented, distinguishing between gradient-based and gradient-free solution schemes. The latter are of more concern regarding the optimisation of industrial problems, i.e. forming processes.

Following this introduction, this thesis is structured as follows:

Chapter 2 introduces basic preliminaries of this work, like the mathematical notation. Additionally, the computational environment, regarding software and hardware utilised in this thesis, is described.

Chapter 3 briefly introduces mathematical optimisation. An optimisation problem, with and without constraints, is formulated and solution strategies are outlined. Specifically, the gradient-free Nelder-Mead simplex, and the gradient-based Sequential Quadratic Programming are presented. Finally, an overview regarding design choices and its application in this thesis is shown.

Part I: Academic challenges encompasses chapters 4-7 and describes the academic challenges of numerical damage optimisation. A nonlocal continuum ductile damage model is presented and enhanced regarding sensitivity data for optimisation problems. To derive the aforementioned gradient information, an efficient method is required, as the underlying material model is already numerically very taxing. A variational approach is chosen to perform sensitivity analysis. The derived gradients are discretised and efficiently implemented into the computational environment and enable gradient-based optimisation.

Chapter 4 summarises the fundamentals of continuum mechanics which are necessary for the conducted research. An introduction to continuum damage modelling is given and solution strategies to solve the ill-posed problem of local damage models are presented. A ductile damage model, using a micromorphic approach to regularise the damage, is presented. The continuous equations are discretised using a FE approach and other aspects of the numerical implementation are depicted.

Chapter 5 enhances the material model by deriving gradients with respect to different design variables. By means of a variational approach, sensitivity analysis is applied in an efficient mathematical framework. Discretisation allows embedding in the numerical FE framework. Critical aspects regarding this numerical implementation are presented and the results from the analytically derived gradients are compared to numerically computed gradient data.

Chapter 6 applies the derived gradients to selected academic optimisation problems. Damage tolerant structures can be generated and the benefits regarding damage modelling within optimisation is highlighted. Additionally, the material sensitivities are used outside an optimisation environment to analyse identifiability of material parameters in a prototype material sensitivity study.

Chapter 7 concludes Part I, i.e. the academic challenges, of this work. The theory and results regarding damage optimisation by means of analytically derived gradients are discussed. Additionally, the transfer to Part II of this thesis is briefly motivated.

Part II: Industrial challenges encompasses chapters 8-11 and describes the methodology utilised to optimise forming processes. The main challenge in this regard are the inherent discontinuous contact mechanics, which get introduced. This has to be considered and therefore utilisation of the concept of the previous part gets very challenging. An optimisation framework is presented, which is able to optimise forming processes and consider the discontinuous problems which arise. Furthermore, the processes have to be directly optimised to predict the damage behaviour of the underlying process as accurately as possible, which enhances the complexity even further.

Chapter 8 provides a background and the reasoning for the optimisation of forming processes. A brief introduction to the forming processes in this thesis, and the current state of the art regarding damage in these fields, is presented. Possible approaches to numerical optimisation, and the chosen approach in this thesis, are discussed.

Chapter 9 presents the created general optimisation framework. It uses the commercial software Abaqus as the FE solver, which is briefly described and nomenclature regarding the software is outlined. The framework, and possible alternative approaches, are described. A optimisation problem is shown, which highlights the applicability of the framework by considering many different types of design variables simultaneously.

Chapter 10 provides results from numerical damage optimisation for three different forming processes. Rod extrusion, bending and deep-drawing are optimised using the previously presented optimisation framework. Optimal process parameters in the fields of bulk metal and sheet metal forming can be generated, which in turn reduce damage accumulation. The framework, depending on the choice of optimiser, enables optimisation of processes with few and many contact interactions all the same.

Chapter 11 concludes Part II, i.e. the industrial challenges, of this work. The key aspects regarding forming, the optimisation framework, as well as important statements due to the results are again highlighted upon.

Chapter 12 finally concludes the complete thesis. The highlights regarding the academic challenges of Part I and the industrial challenges of Part II are summarised. Additionally, based on this work, an outlook on further research with the described methodology is given.

In addition, an appendix is added to this work. Therein, precise equations which were omitted during derivation are depicted. This helps with the readability of this thesis, as the respective sections only denote the most important equations and are not cluttered with many additional derivatives. Furthermore, further results from other optimisation problems are added.

Appendix A derives necessary quantities like the material tangent operator and the local Jacobi matrix for ductile damage material model. Additionally, the utilised $\bar{\boldsymbol{F}}$-method and remarks regarding implementation are briefly summarised.

Appendix B presents the detailed derivation of the analytically derived gradients for the sensitivity analysis of the ductile damage model.

Appendix C includes additional results from Part I: Academic challenges, such as experimental comparison of 3d-printed samples from damage optimised geometries, a parameter identification for the material model used in this work, and raw data from the response sensitivity study.

Appendix D consists of additional data from the optimised forming processes, i.e. rod extrusion, bending and stretch indenting.

Chapter 2

Preliminaries and notations

This chapter presents some preliminaries for this thesis. Different notations for tensors and matrices, as well as special tensor products, are introduced. Additionally, an overview regarding the created and utilised optimisation frameworks and the employed software and hardware are presented.

2.1 Notation

In this thesis, vectors or tensors of first order are denoted as lower-case letters with bold face, italic font, tensors of second order are denoted as upper-case letters with bold face, italic font, and tensors of fourth order are denoted as upper-case letters with blackboard-bold font, e.g.

$$\begin{aligned} \boldsymbol{a} &= a_i\,\boldsymbol{e}_i && (\textit{first-order tensor, vector}),\\ \boldsymbol{A} &= A_{ij}\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j && (\textit{second order tensor}),\\ \mathbb{A} &= A_{ijkl}\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l && (\textit{fourth-order tensor}), \end{aligned} \tag{2.1}$$

with the fixed orthonormal basis $\boldsymbol{e}_{\{1,2,3\}}$. A single contraction is denoted by a dot product, e.g.

$$\boldsymbol{a} \cdot \boldsymbol{b} = a_i\,b_i, \quad \boldsymbol{A} \cdot \boldsymbol{b} = A_{ij}\,b_j\,\boldsymbol{e}_i, \tag{2.2}$$

while double contractions are donated with a colon, e.g.

$$\boldsymbol{A} : \boldsymbol{B} = A_{ij}\,B_{ij}, \quad \mathbb{A} : \boldsymbol{B} = A_{ijkl}\,B_{kl}\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{2.3}$$

In addition, two special dyadic products for tensors of second order are defined as

$$\boldsymbol{A} \,\overline{\otimes}\, \boldsymbol{B} = A_{ik}\,B_{jl}\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l, \tag{2.4}$$

$$\boldsymbol{A} \,\underline{\otimes}\, \boldsymbol{B} = A_{il}\,B_{jk}\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l, \tag{2.5}$$

as well as the special dyadic product for a tensor of second order and a vector, i.e.

$$[\boldsymbol{A} \overline{\otimes} \boldsymbol{a}] \cdot \boldsymbol{b} = [A_{ik}\, a_j]\, b_k\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{2.6}$$

Discrete vectors and matrices are denoted in bold face Roman letters, i.e.

$$\begin{aligned} \mathbf{a} &= [a]_i \in \mathbb{R}^n && (\mathit{vector}), \\ \mathbf{A} &= [A]_{ij} \in \mathbb{R}^{n\times m} && (\mathit{matrix}). \end{aligned} \tag{2.7}$$

Transposed matrices and tensors are denoted with a superscript t, and their inverse with a superscript -1. A transposed inverse is denoted with at superscript $-t$. For a second-order tensor (2.1) this yields

$$\begin{aligned} \boldsymbol{A}^t &= A_{ji}\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j = A_{ij}\, \boldsymbol{e}_j \otimes \boldsymbol{e}_i \\ \boldsymbol{A}^{-1} &= A^{-1}_{ij}\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j \\ \boldsymbol{A}^{-t} &= A^{-1}_{ji}\, \boldsymbol{e}_i \otimes \boldsymbol{e}_j = A^{-1}_{ij}\, \boldsymbol{e}_j \otimes \boldsymbol{e}_i. \end{aligned} \tag{2.8}$$

This notation has been introduced in [99] as it allows a concise presentation of large and otherwise cumbersome formulae. Utilising this notations allows the formulae in this thesis to be rewritten without explicitly following index notation in their respective equations and eases the readability of the presented work.

2.2 Variations and derivatives

In structural analysis, a boundary value problem (BVP) is generally dependent on a set of state or field variables $\boldsymbol{w}$. In the standard elastic case, these are typically the displacements $\boldsymbol{u}$. For more complex material models, for example nonlocal damage models, it is possible to extend this with additional variables, such that the set of field variables consists of more than one set of variables, which is captured in the variables $\boldsymbol{w}$. Furthermore, material behaviour can be dependent on the loading history, due to prior evolution of damage or plasticity for example. This introduces history or state dependent variables $\boldsymbol{h}$ to the functional. Finally, in the case of structural optimisation (SO), the referential configuration no longer remains fixed, but may change throughout the optimisation. This change in referential design is captured by the design variables $\boldsymbol{s}$. Ultimately, this allows the definition of a three field functional $\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h})$. Each respective argument is defined on an appropriate function space, i.e. $\boldsymbol{w} \in \mathcal{W}$, $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{h} \in \mathcal{H}$. Generally, the functional is nonlinear in each of the respective arguments. Further remarks on the mathematical structure are beyond the scope of this thesis, see e.g. [50, 51, 208] for further information.

In this work, the total variations of a functional is denoted with a δ symbol. The total variation of the above functional, assuming it is differentiable, reads

$$\delta\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h}) := \delta_{\boldsymbol{w}}\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h};\delta\boldsymbol{w}) + \delta_{\boldsymbol{s}}\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h};\delta\boldsymbol{s}) + \delta_{\boldsymbol{h}}\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h};\delta\boldsymbol{h}), \tag{2.9}$$

where the subscripts in the variational symbol on the right side of the equation denote the partial variation w.r.t. their respective variable. The partial variations are defined as

$$
\begin{aligned}
\delta_{\boldsymbol{w}}\mathfrak{J} &:= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[\mathfrak{J}(\boldsymbol{w}+\epsilon\,\boldsymbol{\eta},\boldsymbol{s},\boldsymbol{h})-\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h})\right] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathfrak{J}(\boldsymbol{w}+\epsilon\,\boldsymbol{\eta},\boldsymbol{s},\boldsymbol{h})\right|_{\epsilon=0}\\
\delta_{\boldsymbol{s}}\mathfrak{J} &:= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[\mathfrak{J}(\boldsymbol{w},\boldsymbol{s}+\epsilon\,\boldsymbol{\kappa},\boldsymbol{h})-\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h})\right] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathfrak{J}(\boldsymbol{w},\boldsymbol{s}+\epsilon\,\boldsymbol{\kappa},\boldsymbol{h})\right|_{\epsilon=0}\\
\delta_{\boldsymbol{h}}\mathfrak{J} &:= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h}+\epsilon\,\boldsymbol{\mu})-\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h})\right] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathfrak{J}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h}+\epsilon\,\boldsymbol{\mu})\right|_{\epsilon=0}
\end{aligned}
\tag{2.10}
$$

with the test functions $\boldsymbol{\eta}\in\mathcal{W}$, $\boldsymbol{\kappa}\in\mathcal{S}$ and $\boldsymbol{\mu}\in\mathcal{H}$. The variations of the three arguments of the three field functional are thus defined as $\delta\boldsymbol{w}=\epsilon\,\boldsymbol{\eta}$, $\delta\boldsymbol{s}=\epsilon\,\boldsymbol{\kappa}$ and $\delta\boldsymbol{h}=\epsilon\,\boldsymbol{\mu}$. The above derivative is also known as the Gâteaux or directional derivative. The resulting partial variations are linear in their respective direction, while remaining nonlinear w.r.t. the three initial arguments. This is denoted by the separation with the semicolon in (2.9). Arguments on the left side of the semicolon are nonlinear, while linear on the right. These are called semilinear forms. If the arguments are fixed, i.e $\left\{\hat{\boldsymbol{w}},\hat{\boldsymbol{s}},\hat{\boldsymbol{h}}\right\}$, the resulting partial variation yields a bilinear form. For more details on the described concept and principals of variational calculus, see cf. [34, 85, 129, 149, 155].

Remark 2.1 Here the quantity $\mathfrak{J}$ is introduced. The choice of a fractal symbol, instead of the symbol J, is made to distinguish between objectives and the determinants J arising from the kinematics in Chap. 4 and Chap. 5, as well as the Jacobian matrix $\mathbf{J}$ for the local material model in Appendix A.

2.3 Computational environment

2.3.1 Software environment

In this work, a lot of different software packages and programming environments are utilised to generate code, compute simulations and visualise data. The main software regarding the numerical implementation of the presented optimisation problems is achieved using Matlab. Two optimisation frameworks are created and used within this thesis.

AOPS. The first framework, the **A**cademic **O**ptimisation **P**roblem **S**olver, is applied to SO, parameter identification, as well as response sensitivity study. This framework consists of a self-programmed finite element (FE) environment, which allows incorporation of own material models and element formulations in Matlab. These element formulations utilise precompiled `.mex`-functions to speed up computation time, which is further increased by parallelising the global element assembly. The nonlocal ductile damage model is implemented in this framework. The FE environment has to account for the deformation field and nonlocal damage as global field variables. Therefore, it is structured to solve single or multi-field problems, taking the respective boundary conditions into consideration. Furthermore, an automatic time step control is implemented, required due to the complexity of the considered material model. This method automatically adjusts the global pseudo-time steps, if local iterations fail. With lower pseudo-time steps, the global load steps are reduced as well, aiming for the local material model to converge. This code uses analytically derived sensitivity information for gradient-based optimisation schemes. The resulting equations from a sensitivity analysis are discretised and incorporated into the framework to enable efficient computation and hence optimisation problems. Special treatment is required to incorporate the history dependent sensitivity information.

IOPS. The second framework, the **I**undustrial **O**ptimisation **P**roblem **S**olver, exchanges the manually adapted FE environment and computation of sensitivity information from the AOPS, which accounts for the ductile damage behaviour, with the commercial FE software Abaqus. By adjusting the framework to incorporate an external FE solver, more variance in material models and BVPs can be considered in the simulations, due to the more robust implementation and addition of features. This in turn extends to the optimisation problems. Furthermore, adjustments to the input routine have to be made, as the files require drastically different input parameters. Since the gradients, if gradient-based methods are used, within this framework are computed using the finite difference method, these have to be computed and provided as well. While Matlab controls the overall optimisation, the data in the FE solver has to be accessed which is achieved with Python scripting. The specifics are detailed thoroughly in Chap. 9.

An overview for the specifically utilised software environments of both these frameworks is presented in the following.

Matlab. Matlab [156] is a commercial high-level language provided by MathWorks©, with its own numeric computing environment. In contrast to other compiler-based languages, such as C,C++ or Fortran, Matlab is an interpreter based language. Therefore, the code does not need to be compiled and is interpreted during runtime. However, this generally leads to a slower computation time. Even though compilation of the code is not required, it is still possible to convert uncompiled Matlab code into precompiled C or C++ code by using the *Matlab Coder* toolbox. This is especially useful if certain functions are called repeatedly, such as element types in FE simulations, as this can significantly speed up computation times. Additionally, using the *Parallel Computing* toolbox allows further enhancement to the code by enabling parallel computing. This is applied to the FE assemblies within in the first optimisation framework.

Finally, with respect to optimisation, the *Optimisation* toolbox is of great use, as it provides a multitude of solution strategies and solvers for optimisation problems. In this work, the emphasis lies on *fmincon* for gradient-based, and *fminsearch* for gradient-free solution strategies, see Chap. 3. For the control of external software, such as Abaqus or Python scripting, the *system* command is used. An overview of the utilised toolboxes provided by Matlab, and their used functions, is listed in Table 2.1.

Table 2.1: Utilised Matlab toolboxes.

Toolbox	Version	Function
Matlab Coder	5.4	*codegen*
Parallel Computing Toolbox	7.6	*parfor*
Optimization Toolbox	9.3	*fminsearch*, *fmincon*

Paraview. The open-source software ParaView [1] is used to analyse data and visualise the results. By generating `.vtk`-files within Matlab, the results from the structural analysis within each optimisation step is stored in a post-processing step of both frameworks. Beside the FE mesh, these include data at integration point level, such as stresses, strains and history variables, as well as nodal data, like displacements or nonlocal damage. Under specific circumstances, additional data can be stored, to highlight different sections or material properties. This is used to visualise the results for the generated optimal solutions in all presented studies within this thesis.

Abaqus. The commercial software Abaqus [58], distributed by Dassault Systèmes Simulia Corp., is a FE and computer aided engineering (CAE) software. That is, it allows modelling of the process which is later on submitted to its solver. Thereby, modelling of a problem with additional CAE software can be avoided. Due to its general applicability with its many kinds of solvers and material models, it is widely used in academia as well as industry. Its application within optimisation of forming processes in this work allows consideration of contact mechanics within an optimisation framework. The software is programmed in Fortran and Python, and with the latter enables interfacing with the software by Python scripting, to extract and inject data from or into the simulation, respectively.

Python. The general-purpose programming language Python [226] is well-established and, similar to Matlab, does not require compilation of its code. It is most known for its emphasis on code readability by including significant code indention into its implementation. Due to its popularity, especially nowadays in the fields of machine-learning, a lot of libraries for special tasks are available to load and use. In the field of mathematics, *NumPy* [107] and *SciPy* [228] are necessary libraries to ease with computation of formulae and their visualisation. In addition, the libraries provided by Abaqus allow interfacing with the simulation software and thereby are necessary for the evaluation of simulation data during the automated optimisation. In this work, due to limitations with Abaqus, Python 2.7 has to be used.

Blender. The open-source software Blender [55] is generally used to render 3d computer graphics for animations or visual effects. In this work however, it is used to create 3d samples of structural optimised geometries for 3d printing and the validation of the numerical results. The Computer Aided Geometric Design data from the optimisation is loaded and adapted to create `.stl`-files for the 3d-printing software.

2.3.2 Hardware environment

The results in this thesis are achieved within multiple hardware environments, see Table 2.2. The respective operating system is annotated in the description column.

Table 2.2: Hardware environments.

Machine label	Description
DWS1	Dell Mobile Precision Workstation 7520 (64 bit Linux OS) Intel© Core™ i7-7920HQ (quad-core @ 3.1 GHz (4.1 GHz boost)) 32 GB DDR4 Memory
DWS2	Dell Mobile Precision Workstation 5570 (64 bit Linux OS) Intel© Core™ i7-12800H (Tetradeca-core @ 2.4 GHz (4.8 GHz boost)) 32 GB DDR4 Memory
LiDO3	Linux Cluster Dortmund (64 bit Linux OS) 2x Intel© Xeon© E5 2690v4 (Tetradeca-core @ 2.6 GHz base (3.5 GHz boost)) up to 256 GB DDR4 Memory
DPC	Desktop Personal Computer (64 bit Windows 10) AMD Ryzen™ 7 3700x (Octa-core @ 3.6 GHz (4.4 GHz boost)) 16 GB DDR4 Memory

Chapter 3

Optimisation setup overview

In this chapter, the general concept of mathematical optimisation is presented. The optimisation problems with and without constraints are formulated and solution strategies to solve such problems are briefly explained. An overview of the numerical treatment and the computation of gradient information is given. Finally, the optimisation problems, their denomination and possible design variables as they arise in this work are categorised and listed.

3.1 Mathematical optimisation

In this chapter, a general overview regarding mathematical optimisation is presented. In mathematical optimisation a scalar valued function is defined as an objective and subjected to a minimisation problem. This objective depends on a choice of design variables. Subsequent changes in these variables lead to changes in the objective value and are the arguments of the mathematical optimisation problem. Since the concept of mathematical optimisation is not restrained to e.g. structural optimisation, the algorithms presented later on can be used for any stated optimisation problem. Furthermore, in this work the term numerical optimisation is used to describe mathematical optimisation problems which are solved within a numerical framework and solved using numerical algorithms and solution strategies.

In mathematical optimisation, many different distinctions are made regarding the type of optimisation problem at hand, such as

- constrained vs. unconstrained problem,
- convex vs. concave problem,
- smooth vs. nonsmooth problem,
- linear vs. nonlinear problem,
- few vs. many design variables,

to name a few. Depending on the type of optimisation problem, the choice of solution strategy has to be specifically chosen. This choice greatly effects the outcome of the solution, or in extreme cases the general convergence rate of the chosen algorithm. For example, gradient-based methods struggle with nonsmooth problems since the gradient definition changes depending on the direction in which the discontinuity within the optimisation problem is approached.

In this work the focus is placed on gradient-based and gradient-free methods, and some insight is provided. However, as already described, the field of mathematical optimisation is vast and it is not feasible to present all possible algorithms here. A reader is therefore referred to standard literature, cf. e.g. [113, 174, 194], for an overview of solution techniques and their respective fields of optimisation.

3.2 Mathematical optimisation problem formulation

To define an optimisation problem, let $\mathfrak{J}$ be a scalar valued objective function which depends on a specifically defined choice of design variables $\boldsymbol{s} \in \mathbb{R}^n$. An unconstrained optimisation problem reads

$$\underset{\boldsymbol{s}\in\mathbb{R}^n}{\text{minimise}} \quad \mathfrak{J}(\boldsymbol{s}). \tag{3.1}$$

This problem can be restricted by boxed constraints in the form of upper $\boldsymbol{s}_\mathrm{u}$ and lower $\boldsymbol{s}_\mathrm{l}$ bounds, which are applied to the design variables. Furthermore, constraints in the form of (linear or nonlinear) equality constrains $\boldsymbol{h} = \boldsymbol{0}$ and inequality constrains $\boldsymbol{g} \leq \boldsymbol{0}$ can be applied to the problem. A general constrained problem thus reads

$$\begin{aligned} \underset{\boldsymbol{s}\in\mathbb{R}^n}{\text{minimise}} \quad & \mathfrak{J}(\boldsymbol{s}) \\ \text{subject to} \quad & \boldsymbol{h}(\boldsymbol{s}) = \boldsymbol{0} \\ & \boldsymbol{g}(\boldsymbol{s}) \leq \boldsymbol{0} \\ & \boldsymbol{s}_\mathrm{l} \leq \boldsymbol{s} \leq \boldsymbol{s}_\mathrm{u}. \end{aligned} \tag{3.2}$$

To solve problems (3.1) and (3.2), iterative procedures are required. These aim to find a solution, often based on an initial guess $\boldsymbol{s}_0$, in the defined design space according to the chosen solution strategy. These are discussed in Sect. 3.3 with the focus placed on the applied methods used in this thesis.

3.3 Solution strategies

As mentioned previously, the choice of solution strategy depends strongly on the type of optimisation problem at hand. Herein, the focus is placed on the distinction between gradient-free and gradient-based optimisation strategies, as both of these topics are of vital concern regarding this thesis. Furthermore, the specific emphasis is put on the Nelder-Mead simplex for the gradient-free optimisation, and Sequential Quadratic Programming for the gradient-based optimisation. Both of these algorithms are used within this work.

Another aspect to consider is the distinction between local and global solutions. Generally, a solution generated by optimisers for nonlinear problems yield local solutions. A global solution can only be guaranteed in special cases, such as for linear, convex problems where the local solution is always the global one. Therefore, it is generally advisable to use, if possible, multiple initial guesses $\boldsymbol{s}_0$ in any optimisation. Depending on the type of problem, that is not always possible. Therein, deflation algorithms, cf. [49, 178, 179], can be used to generate alternative solutions based the same initial guess.

3.3.1 Gradient-free optimisation

Many methods within the field of gradient-free optimisation exist. Since these methods only evaluate the objective $\mathfrak{J}$, no requirements regarding convexity or smoothness are required. Therefore, gradient-free methods can generally be applied to any unconstrained optimisation problem (3.1). Application of these methods to constrained problems (3.2) is not straightforward and is discussed later. The main drawback for these methods is the computational cost due to a high number of function evaluations. Therefore, application of gradient-free methods is generally only advisable to problems with few design variables. Simple procedures like grid-search methods span an even spaced grid over the design space and evaluate the objective at intersections of the grid [29]. Another well-known approach is the Monte-Carlo method, wherein, instead of an even spaced grid, these sample points are randomly selected throughout the design space [73, 77, 159].

Of specific importance with regard to this thesis is the simplex-reflection method by Nelder and Mead [170]. Notably, this method is not comparable to the simplex technique in linear optimisation problems. This method is available within the optimisation toolbox within Matlab as the function *fminsearch*. The basic concept is to create a simplex for the design, which is moved throughout the design space in an efficient manner to find an optimal solution. A simplex of dimension $n_{\mathrm{dv}} + 1$, where n_{dv} is the number of design variables in the design space, is created from the initial guess $\boldsymbol{s}_0$. This is achieved by evaluating the initial guess and additionally computing the objective value by perturbing each design variable individually. The data from the initial simplex is stored in ascending order, such that

$$\mathfrak{J}(\boldsymbol{s}_1) \leq \mathfrak{J}(\boldsymbol{s}_2) \leq \cdots \leq \mathfrak{J}(\boldsymbol{s}_{n_{\mathrm{dv}}+1}), \tag{3.3}$$

resulting in the corresponding ordering of the design variables $\{\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{n_{\mathrm{dv}}+1}\}$. With this, a reflective centroid is computed, i.e.

$$\bar{\boldsymbol{s}} = \frac{1}{n_{\mathrm{dv}}} \sum_{i=1}^{n_{\mathrm{dv}}} \boldsymbol{s}_i, \tag{3.4}$$

which is used to reflect the worst design, i.e. $\boldsymbol{s}_{n_{\mathrm{dv}}+1}$, using the formula

$$\boldsymbol{s}^{\gamma}(\gamma) = \bar{\boldsymbol{s}} + \gamma \left[\bar{\boldsymbol{s}} - \boldsymbol{s}_{n_{\mathrm{dv}}+1}\right]. \tag{3.5}$$

Depending on the criteria met, the scaling parameter γ takes on different values, i.e. $\gamma = 1$ in the reflective case, $\gamma = 2$ in the expansive case and $\gamma = \pm 0.5$ for the outside contraction or inside contraction, respectively. If neither of these new points generate an improvement over the function value $\mathfrak{J}(\boldsymbol{s}_{n_{\mathrm{dv}}+1})$, the simplex is shrunk, while keeping the current optimal point $\boldsymbol{s}_1$ fixed. This remaining new design variables are computed via

$$\boldsymbol{s}_j = \boldsymbol{s}_1 + \gamma(\boldsymbol{s}_j - \boldsymbol{s}_1), \quad \text{for} \quad j = 2,3,\dots,n_{\text{dv}} + 1. \tag{3.6}$$

The method is sketched in Algorithm 3.1 and an update visualised in Fig. 3.1 for $n_{\text{dv}} = 2$.

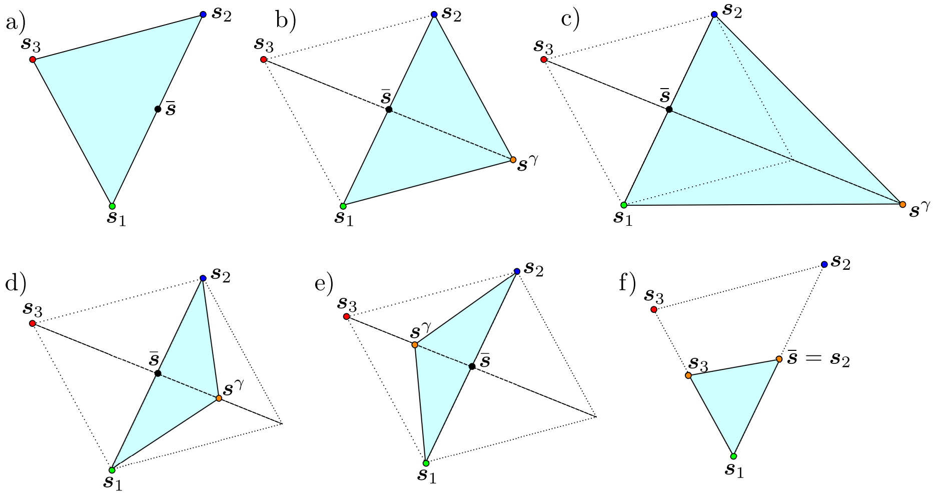


Figure 3.1: Graphical illustration of the Nelder-Mead simplex procedures according to Algorithm 3.1: a) Current simplex, b) reflection, c) expansion, d) outside contraction, e) inside contraction, f) shrinkage. The blue area depicts the simplex after each procedure.

Another drawback of gradient-free optimisation is the consideration of constraints, of either boxed, equal or unequal variety (3.2). A simple strategy for the previously described method is to penalise the objective $\mathfrak{J}(\boldsymbol{s})$ of (3.1), if constraints are violated, e.g.

$$\underset{\boldsymbol{s}\in\mathbb{R}^n}{\text{minimise}} \quad \mathfrak{J}^{\text{pen}}(\boldsymbol{s}) = \mathfrak{J}(\boldsymbol{s}) + \sum_{i=1}^{n_{\text{g}}} \beta_g \left[|g_i(\boldsymbol{s})| + g_i(\boldsymbol{s})\right]^2 + \sum_{j=1}^{n_{\text{h}}} \frac{\beta_h}{2} \left[h_j(\boldsymbol{s})\right]^2 , \tag{3.7}$$

where n_{g} is the number of inequality constraints and n_{h} the number of equality constraints. The penalty parameters $\beta_\bullet$ are chosen such that the violation of constraints significantly affect the objective $\mathfrak{J}$. This approach however has some significant drawbacks. On the one hand, the penalisation is only active if constraints are violated. Therefore, solutions will most likely result in a small violation of a constraint, since an optimal solution is approached from the direction, where a constraint is violated. This in turn might lead to convergence problems for the underlying numerical simulation, since those constraints might be applied due instabilities that may occur if they are exceed too much. On the other hand, applying many constraints might lead the optimiser to never converge with in the admissible space. Therefore, application of a penalty method is only advisable when only a few constraints are considered simultaneously. Inclusion of boxed constraints can be handled more efficiently. Applying a quadratic or $\sin(\boldsymbol{s})$ transformation to the design variables, forces them to always be considered in the admissible space, cf. [57].

Algorithm 3.1 The Nelder-Mead simplex algorithm.

```
Sort functions values: $\mathfrak{J}(\boldsymbol{s}_0) \leq \mathfrak{J}(\boldsymbol{s}_1) \leq \cdots \leq \mathfrak{J}(\boldsymbol{s}_{n_{\mathrm{dv}}+1})$
Sort design variables according to function values: $\{\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{n_{\mathrm{dv}}+1}\}$
$\gamma \leftarrow 1$, compute $\boldsymbol{s}^{\gamma}(1)$, compute $\mathfrak{J}(\boldsymbol{s}^{\gamma}(1))$
//Evaluate reflective solution
if $\mathfrak{J}(\boldsymbol{s}^{\gamma}) \leq \mathfrak{J}(\boldsymbol{s}_1)$ then
    //Perform expansion
    $\gamma \leftarrow 2$, compute $\boldsymbol{s}^{\gamma}(2)$ compute $\mathfrak{J}(\boldsymbol{s}^{\gamma}(2))$
    if $\mathfrak{J}(\boldsymbol{s}^{\gamma}(2)) \leq \mathfrak{J}(\boldsymbol{s}^{\gamma}(1))$ then
        $\boldsymbol{s}_{n_{\mathrm{dv}}+1} \leftarrow \boldsymbol{s}^{\gamma}(2)$
        → go to next iteration
    else
        $\boldsymbol{s}_{n_{\mathrm{dv}}+1} \leftarrow \boldsymbol{s}^{\gamma}(1)$
        → go to next iteration
    end if
else
    if $\mathfrak{J}(\boldsymbol{s}^{\gamma}(1)) \leq \mathfrak{J}(\boldsymbol{s}_{n_{\mathrm{dv}}})$ then
        $\boldsymbol{s}_{n_{\mathrm{dv}}+1} \leftarrow \boldsymbol{s}^{\gamma}(1)$
        → go to next iteration
    else
        //Perform contractions
        if $\mathfrak{J}(\boldsymbol{s}^{\gamma}(1)) \leq \mathfrak{J}(\boldsymbol{s}_{n_{\mathrm{dv}}+1})$ then
            //Outside contraction
            $\gamma \leftarrow 0.5$, compute $\boldsymbol{s}^{\gamma}(0.5)$, compute $\mathfrak{J}(\boldsymbol{s}^{\gamma}(0.5))$
            if $\mathfrak{J}(\boldsymbol{s}^{\gamma}(0.5)) \leq \mathfrak{J}(\boldsymbol{s}^{\gamma}(1))$ then
                $\boldsymbol{s}_{n_{\mathrm{dv}}+1} \leftarrow \boldsymbol{s}^{\gamma}(0.5)$
                → go to next iteration
            else
                Perform a shrink
            end if
        else
            //Inside contraction
            $\gamma \leftarrow -0.5$, compute $\boldsymbol{s}^{\gamma}(-0.5)$, compute $\mathfrak{J}(\boldsymbol{s}^{\gamma}(-0.5))$
            if $\mathfrak{J}(\boldsymbol{s}^{\gamma}(-0.5)) \leq \mathfrak{J}(\boldsymbol{s}^{\gamma}(1))$ then
                $\boldsymbol{s}_{n_{\mathrm{dv}}+1} \leftarrow \boldsymbol{s}^{\gamma}(-0.5)$
                → go to next iteration
            else
                Perform shrinkage
            end if
        end if
        //Shrinkage
        $\gamma \leftarrow 0.5$
        $\boldsymbol{s}_j = \boldsymbol{s}_1 + \gamma(\boldsymbol{s}_j - \boldsymbol{s}_1), \quad j = 2,3,\ldots,n_{\mathrm{dv}}+1$
        → go to next iteration
    end if
end if
```

3.3.2 Gradient-based optimisation

Gradient-based methods consist of the most efficient optimisation strategies available and are often recommended, if possible, to be used over gradient-free methods. Due to the utilisation of additional information by computing the gradient of the objective and constraints, this can significantly speed up the iterative process and yield results in fewer iterations. However, gradient-based methods require the considered functions, i.e. $\mathfrak{J}(\boldsymbol{s}), \boldsymbol{h}(\boldsymbol{s})$ and $\boldsymbol{g}(\boldsymbol{s})$, to be continuous in $\boldsymbol{s}$ up to the order of chosen method.

Due to the efficiency of gradient-based methods, a lot of research is done in this field leading to many specialised solution strategies for specific cases. The Levenberg-Marquadt method for example is specifically designed to solve least-squared problems [143, 154]. Other methods such as the interior-point method [81, 92] nowadays sees a lot of research and application due to its efficient handling of constraints. The method applies a barrier function to the constraints to solve an unconstrained problem. For a decreasing value in the barrier parameter $\mu \to 0$, the problem reads

$$\begin{aligned} \underset{\boldsymbol{s}\in\mathbb{R}^n}{\text{minimise}} \quad & \mathfrak{J}^{\text{ip}}(\boldsymbol{s}) = \mathfrak{J}(\boldsymbol{s}) - \mu \sum_{i=1}^{n_\text{g}} \log(c_i(\boldsymbol{s})) \\ \text{subject to} \quad & \boldsymbol{h}(\boldsymbol{s}) = \boldsymbol{0} \\ & \boldsymbol{g}(\boldsymbol{s}) + \boldsymbol{c} = \boldsymbol{0}, \end{aligned} \tag{3.8}$$

where c_i are the slack variables for inequality constraints g_i. The logarithmic barrier term prevents the slack variables from reaching negative values and therefore fulfils the additional slack condition $\boldsymbol{c} \geq \boldsymbol{0}$ implicitly. The huge benefit of this method is the approach of the solution from the admissible interior space within each iteration. It is therefore very useful if constraint violations lead to convergence problems, such as in crystal plasticity for example [173], where a highly constrained optimisation problem has to be considered. Due to its efficiency it also sees a lot of application topology and shape optimisation [116, 189], and is available in many software like the Matlab toolbox *fmincon* or open source tools like *Ipopt* [229] for the Python environment. The gradient-based method of consideration for this thesis however is Sequential Quadratic Programming (SQP), which will be briefly outlined here.

Sequential Quadratic Programming is a very effective method to solve constrained optimisation problems and outperforms other tested methods in nonlinear programming [194]. The basic idea is to enhance Newton's method to take equality and inequality constraints into consideration. Initially the problem is rewritten into the Lagrange function $\mathcal{L}$, i.e.

$$\begin{aligned} \mathcal{L}(\boldsymbol{s},\lambda,\mu,c) &:= \mathfrak{J}(\boldsymbol{s}) + \sum_{i=1}^{n_\text{h}} \lambda_i h_i(\boldsymbol{s}) + \sum_{j=1}^{n_\text{g}} \mu_j \left[g_j(\boldsymbol{s}) + c_j^2 \right] \\ &= \mathcal{L}(\boldsymbol{s},\boldsymbol{\lambda},\boldsymbol{\mu}) := \mathfrak{J}(\boldsymbol{s}) + \boldsymbol{\lambda}^t \boldsymbol{h}(\boldsymbol{s}) + \boldsymbol{\mu}^t \left[\boldsymbol{g}(\boldsymbol{s}) + \boldsymbol{c} \right], \end{aligned} \tag{3.9}$$

with the Lagrange parameters, also known as the adjoint variables, λ_i and μ_j with $\mu_j \geq 0$, as well as the slack variables c_j. The necessary condition for a minimum enforces the gradient of Lagrange function $\mathfrak{L}$ to fulfil the optimality criteria, i.e.

$$\nabla\mathfrak{L} \overset{!}{=} \mathbf{0}. \tag{3.10}$$

Applying Newton's method, and reordering of the equations, leads to the quadratic subproblem

$$\begin{aligned} \underset{\Delta\boldsymbol{s}}{\text{minimise}} \quad & \tilde{\mathfrak{J}} = \nabla\mathfrak{J}(\boldsymbol{s}^k)\Delta\boldsymbol{s}^k + \frac{1}{2}(\Delta\boldsymbol{s}^k)^t\nabla_{\boldsymbol{ss}}\mathfrak{L}(\boldsymbol{s}^k,\boldsymbol{\lambda}^k\boldsymbol{\mu}^k)\Delta\boldsymbol{s}^k \\ \text{subject to} \quad & \boldsymbol{h}(\boldsymbol{s}^k) + (\nabla\boldsymbol{h}(\boldsymbol{s}^k))^t\Delta\boldsymbol{s}^k = 0 \\ & \boldsymbol{g}(\boldsymbol{s}^k) + (\nabla\boldsymbol{g}(\boldsymbol{s}^k))^t\Delta\boldsymbol{s}^k \leq 0 \\ & \boldsymbol{s}_l - \boldsymbol{s}^k \leq \Delta\boldsymbol{s} \leq \boldsymbol{s}_u - \boldsymbol{s}^k, \end{aligned} \tag{3.11}$$

which is solved in each iterative step to solve the initially stated problem (3.2). The quadratic problem is computed with solution algorithms derived for this specific problem. In this thesis, the function *fmincon* within the Matlab toolbox is used. This implementation uses an active-set algorithm, cf. [79, 174], to solve the quadratic subproblem [156]. For more information on SQP in literature, cf. [36, 183, 194].

Since this method is a gradient-based method of second order, the second derivative of the Lagrange function, and therefore the objective function $\mathfrak{J}$ and constraints $\boldsymbol{h}$ and $\boldsymbol{g}$, is required. Generally, this second derivative is approximated iteratively within a Quasi-Newton method, for example by the Broyden-Fletcher-Goldfarb-Shanno update formula (BFGS) [44, 78, 91, 201]. This method approximates the Hessian, from the initial guess of the identity matrix, and updates it in each subsequent iteration. Notably, this method converges towards the exact analytical solution with each and provides positive definite approximation of the Hessian.

Each solution of the quadratic subproblem yields an update $\Delta\boldsymbol{s}^k$ for the design variables, such that

$$\boldsymbol{s}^{k+1} = \boldsymbol{s}^k + \alpha_k\Delta\boldsymbol{s}^k, \tag{3.12}$$

with α_k as the step-size parameter, which can be optimally chosen by means of a line-search algorithm, cf. [174].

3.4 Numerical treatment

Typically, standard optimisation frameworks follow the same structure. Such a procedure is depicted in Fig. 3.2. Initially, the optimisation problem has to be defined, i.e. either problem (3.1) or (3.2), with an explicit dependency on the design variables $\boldsymbol{s}$. The initialisation of the optimisation generally requires an initial guess $\boldsymbol{s}_0$ as the starting point of the optimisation. Based from this initial guess, the problem is setup, e.g. by Computer Aided Geometric Design (CAGD) in shape optimisation. A structural analysis is computed which returns the resulting structural response due to the initial guess $\boldsymbol{s}_0$. If a gradient-based optimisation is chosen as the solution strategy, the gradient information for the objective and constraints has to be computed. Based on the chosen approach, see

Sect. 3.4.1, this requires additional implementation on how to compute those gradients before starting the optimisation. If a gradient-free method is chosen, this additional step is not necessary and the mathematical optimisation is conducted. This will yield an update $\Delta \boldsymbol{s}$ for the design variables, i.e. a new set of design variables $\boldsymbol{s}_1 = \boldsymbol{s}_0 + \Delta \boldsymbol{s}$. This new set of design variables is used to generate an updated problem for the structural analysis. After each new design iteration, a convergence criterium is checked. If this is passed, the new design is the final optimal design. Otherwise, the process is continued by running a structural analysis based on the new iterative design values.

3.4.1 Acquisition of gradient information

The acquisition of the gradient information for gradient-based optimisation is of key importance regarding the computational effectiveness. Here, two approaches to compute this information are highlighted:

1. Numerically computed gradients by perturbing the design variables,
2. Analytically deriving the gradients and applying discretisation.

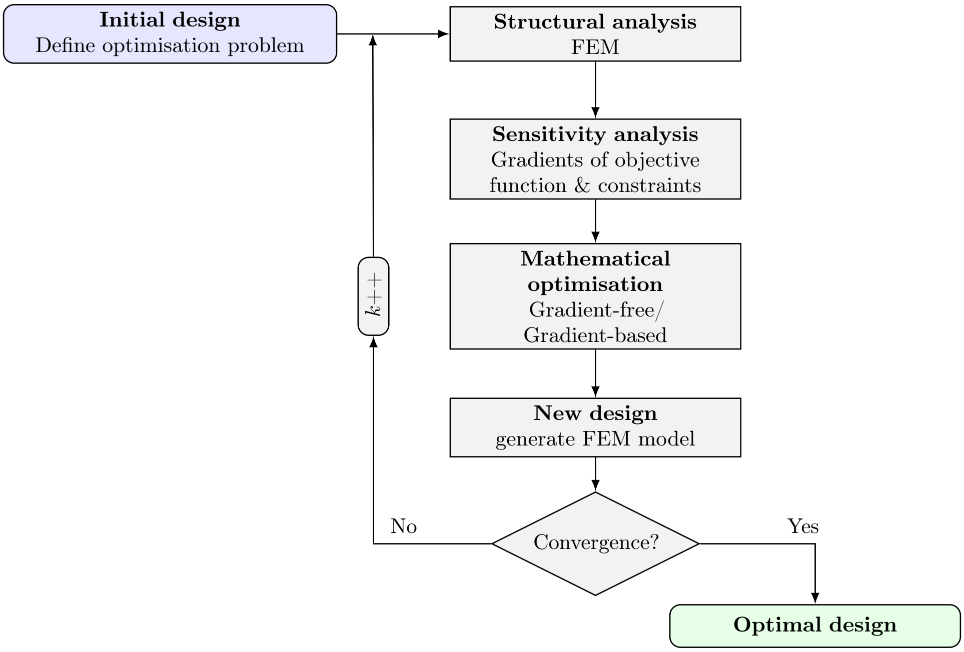

Initial design
Define optimisation problem
Structural analysis
FEM
Sensitivity analysis
Gradients of objective function & constraints
Mathematical optimisation
Gradient-free/ Gradient-based
New design
generate FEM model
k++
Convergence?
No
Yes
Optimal design

Figure 3.2: The general framework of an optimisation procedure. In gradient-free optimisation the sensitivity analysis is omitted.

Finite difference method: The finite difference method (FDM) is a simple approach by numerically perturbing functions and generating numerically approximated gradients. This approach uses a Taylor series approximation

$$\mathfrak{J}(s \pm \epsilon) = \mathfrak{J}(s) \pm \epsilon \, \nabla_s \mathfrak{J}(s) + O(\epsilon^2), \tag{3.13}$$

with perturbation factor $\epsilon \ll 1$. Omitting terms of order 2 and higher $O(\epsilon^2)$, leads to three variants of approximations for the gradient $\nabla_s \mathfrak{J}^h$ using the FDM

$$\begin{aligned}
\nabla_s \mathfrak{J}^h_{\mathrm{FDQ}}(s) &= \frac{\mathfrak{J}(s+\epsilon) - \mathfrak{J}(s)}{\epsilon} && \text{(forward FDM)}, \\
\nabla_s \mathfrak{J}^h_{\mathrm{BDQ}}(s) &= \frac{\mathfrak{J}(s) - \mathfrak{J}(s-\epsilon)}{\epsilon} && \text{(backward FDM)}, \\
\nabla_s \mathfrak{J}^h_{\mathrm{CDQ}}(s) &= \frac{\mathfrak{J}(s+\epsilon) - \mathfrak{J}(s-\epsilon)}{2\epsilon} && \text{(central FDM)},
\end{aligned} \tag{3.14}$$

where the latter FDM is a combination of the two former ones and generally more accurate, while requiring significantly more computational effort. The main benefit of this approach is the easy implementation as only the objective value has to be known and computable. Even though from a mathematical standpoint, the result approaches the analytical gradient for $\epsilon \to 0$, in the numerical environment rounding errors due to computational accuracy will yield a degradation in accuracy for very small values of ϵ. From observation, an optimal perturbation value normally lies in the range of $\epsilon \approx 1 \times 10^{-5} - 1 \times 10^{-8}$, cf. [145]. Another factor to consider is noise within gradients computed with the FDM, cf. [174], which might lead to inaccurate gradient information, for example in nonsmooth or discontinuous problems. In *fmincon* computation of numerical gradients is possible, wherein the perturbation value for example can be defined beforehand. This is used within Part II of this work for optimisation using gradient-based methods. In Part I, the FDM is only used to verify the analytically derived gradients.

In [218] an alternative approach regarding numerically derived gradients was proposed. This approach perturbs the objective with an imaginary value, which leads to significantly more accurate results in the numerically derived gradient information. Similarly to the FDM, using a Taylor-series expansion and negligence of higher order terms finally results in the complex-step derivative approximation (CSDA)

$$\nabla_s \mathfrak{J}^h(s) \approx \frac{\mathrm{Im}\left(\mathfrak{J}(s + i\epsilon)\right)}{\epsilon}, \tag{3.15}$$

with the imaginary number i and $\mathrm{Im}(\bullet)$ as the imaginary part of the function. In contrast to the typical FDM, CSDA yields more accurate results with increasingly smaller perturbation values. Application of this method however, is only possible if the perturbed objective function can treat complex inputs.

Note, that both of these approaches are presented with an exemplary scalar valued design variable s here. Enhancements for multiple design variables $\boldsymbol{s}$ therefore require perturbation of design variable s_i and hence the computations of gradient information $\nabla_{s_i} \mathfrak{J}$. These partial gradients have to be subsequently assembled into the total gradient $\nabla_{\boldsymbol{s}} \mathfrak{J}$. Furthermore, the objective $\mathfrak{J}$ was utilised as an exemplary function or functional.

The same methodology can also be applied to other functions or functionals, e.g. the constraints $\boldsymbol{h}$ and $\boldsymbol{g}$.

Analytical derivation: A more accurate and computationally efficient, albeit complex, approach to deriving gradient information is by means of analytical derivation. This, by hand, derived solution is numerically implemented, allowing computation of the derivative in the chosen numerical environment. Generally, two approaches can be followed to derive these gradients. Either, the discretised functions are used as a basis and sensitivity analysis subsequently applied to these discrete quantities [222]. This method is known as the discrete approach. While this approach is quite straightforward, its numerical efficiency is lacking. The other approach utilises the continuous quantities, e.g. the weak form of equilibrium for the mechanical problem, prior to discretisation. Variation of these continuous functionals allows computation of continuous sensitivity information. For the numerical implementation, the discretisation is applied to these continuous sensitivities. This approach, known as the variational approach, is computationally more efficient, cf. [16, 17, 130], albeit it requires a deep understanding of the underlying continuous equations. As is described in Sect. 1.2, this is the chosen approach and its utilisation for the ductile damage model the main objective of Part I: Academic challenges.

These two approaches find application within this work. Beside the two, other methods exist. The semi-analytical approach couples numerically computed gradients with analytically derived ones, when analytical derivation is not possible for the complete problem. Another important method nowadays, widely used in machine learning [21], is automatic differentiation [169, 207]. This method computes exact gradient informations simultaneously within the function evaluation. However, this type of implementation requires complete redefinition of all used computation formulae to also include gradient calculation. Therefore, inclusion of this concept into a framework has to be done from the ground up and initially be considered. Furthermore, a lot of additional computation time is generated, since each function evaluation also comes with a computation of its derivative. In the case of ductile damage, an already computationally very complex task, this would lead to an enormous increase in computational cost.

3.4.2 Overview of optimisation problems

In this thesis, the concept of mathematical optimisation is applied to multiple types of problems. These are not limited to structural optimisation, as introduced in Chap. 1, but extended to other fields of research within optimisation.

Shape optimisation is a field of structural optimisation, wherein the utilisation of CAGD allows free form optimisation of the general shape of a structure [74, 86]. The control points of the CAGD, or in some cases the nodal coordinates, act as the design variables and their change throughout the optimisation allows a change in general design. Utilisation of CAGD data significantly improves the definition of the optimisation problem, as application of constraints and bounds are easier to define and apply to the problem.

Parameter identification considers optimisations, wherein the constitutive material parameters from the material model are fitted to experimental data, cf. [146, 152, 190,

Table 3.1: An overview for the types of optimisations and their design variables.

Optimisation type	**Design variables**	**Examples**
shape optimisation	control points (CAGD), general geometric description, nodal coordinates	$\mathbf{X}$,$\mathbf{p}$
parameter identification, response sensitivity study	constitutive/material parameters	$\mathbf{m}_{\mathrm{p}}$,E,σ_{y0},c_{d}
process optimisation	process parameters (geometry, friction, forces, displacements,...)	r,α,μ,$\varepsilon_{\mathrm{ex}}$

191]. The objective is typically formulated as a least-square problem between simulated and experimental data. Nowadays with the advancements in technology, more detailed and accurate fittings further enhance this objective by using digital image correlation from the experiments. This allows inclusion of displacement data into the objective, which substantially enhanced the fitted data and therefore the accuracy of the modelling, cf. [187, 192, 215]. Response sensitivity study uses the same data as in a parameter identification, but instead of optimising a set of material parameters, the sensitivity data is directly analysed. This enables statements regarding the efficiency of certain geometries and their effect on identifiability of material parameters.

The term process optimisation in the following describes any optimisation applied to forming processes. These encompass geometric optimisations similar to within shape optimisation, but also allow sizing effects applied tools which are present in the considered forming process. Additionally, boundary conditions like loads or displacements fall under this category. Generally, the design variables of this type of optimisation consider a general set of process parameters which impact the chosen objective or constraint of the optimisation.

For a concise overview, Table 3.1 presents the three discussed types of optimisation, their design variables and some examples for their respective choices, as they will be discussed in this thesis.

Part I:
Academic challenges

Chapter 4

Structural analysis for ductile damage materials

This chapter presents the material model for the underlying structural analysis. The basic kinematics for large deformations and the underlying balance equations are briefly summarised. A mesh-independent ductile damage material model is derived. The damage is regularised by means of a nonlocal micromorphic approach. The local, constitutive model coupling plasticity and damage is depicted. This basis has to be provided for the analytical sensitivity analysis in Chap. 5. Since the chosen methods utilises the continuous equations, they form the foundation of the upcoming work. Finally, all underlying equations are discretised for the finite element implementation and an active-set strategy to solve evolution equations for the local material model is outlined for its implementation into AOPS.

4.1 Continuum damage modelling

To accomplish the objective of this Part I: Academic challenges, as was described in Sect. 1.2, the continuum mechanics ductile damage material model has to be described. This in return requires a lot of basics, which build the foundation for accurate modelling, which is required to perform optimisation. While this information is not necessarily new, it is required to perform the analytical sensitivity analysis in the following chapter, i.e. Chap. 5. The approach therein requires a detailed derivation of all underlying continuous equations. Since the sensitivity analysis is performed prior to discretisation, these equations need to be known and deeply understood. Without this foundation, the sensitivity analysis cannot be performed. Even though this information is not new, the scientific baseline is required as a foundation for the enhancements in Chap. 5. Therefore, these fundamentals have to be provided in this chapter, to be built upon in the following one.

To simulate and predict material behaviour, such as plasticity and damage, the respective dissipative processes have to be accurately modelled. Since the simulated results shall predict actual real-life outcomes, the modelling approaches have to satisfy the balance laws and the thermodynamic laws which govern physics. A differentiation between small deformations or a finite deformation viewpoint leads to different fundamental kinematics. Furthermore, depending on the choice of modelling of microscopic or macroscopic effects, choices regarding accuracy and homogenisation have to be made.

While modelling of plastic effects is well-established, modelling of continuum damage mechanisms is still a field of current research. Its origin traces back to the works conducted by [124], who introduced a scalar variable which coincided with either completely intact material or fully damaged material. Ductile damage, as understood in this work, is the degradation of the macroscopic elastic material properties in metals due to extensive loading. Local defects on the microscale, such as microcracks and voids, start to form, grow, and coalesce, eventually leading to the degradation of the macroscopic behaviour of a given body, cf. [188]. For an overview of the field of continuum damage mechanics and extension to phenomenological effects, cf. e.g. [167]. To capture these degradation effects, typically a $[1-d]$ approach is utilised, established by [141, 142], which has been applied in many following modelling methods, e.g. cf. [41, 80, 230]. Another well-known material model, especially in the context of ductile damage, is the work by Gurson [105] which is motivated by a micromechanical void volume fraction. This model was extended by Tvergaard and Needleman [225] and is known as the Gurson–Tvergaard–Needleman (GTN) model, which is widely used in the simulation of damage processes within industrial applications. In general, damage is an anisotropic process. To capture this, a tensor valued function can be introduced to define separate damage evolutions in all three dimensions. However, in this work damage is simplified and modelled for an isotropic case. For more information on non-regularised and regularised anisotropic damage, cf. [32, 48, 70, 158] and [60, 122, 214], respectively. The material model used within this work has been derived in [215], which couples plasticity with damage to describe the ductile damage behaviour of the materials considered in this thesis.

To capture the modelling aspect in numerics, the equations have to be discretised. In this work, the finite element method (FEM), a well-established form of simulation environment, is used. FEM has been a tool to solve complex numerical problems for many years and a lot of textbooks exist, which elaborate the concept, cf. e.g. [37, 120, 238].

4.2 Kinematics

Consider a referential body configuration $\mathcal{B}_0 \subset \mathbb{R}^3$ in its initial, undeformed state in the three-dimensional setting. Any point $\boldsymbol{X} \in \mathcal{B}_0$ in this referential position can be mapped to the point $\boldsymbol{x} \in \mathcal{B}_t$ in the deformed, current configuration $\mathcal{B}_t \subset \mathbb{R}^3$ at time $t \in [0,T]$. The nonlinear mapping $\boldsymbol{\varphi}$, see Fig. 4.1, is defined as

$$\boldsymbol{\varphi} : \begin{cases} \mathcal{B}_0 \longrightarrow \mathcal{B}_t \\ (\boldsymbol{X},t) \mapsto \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X},t) \end{cases} . \tag{4.1}$$

Given an arbitrary differentiable function, the gradient and divergence operators w.r.t. the reference configuration can be defined as

$$\operatorname{Grad}(\bullet) = \nabla_{\boldsymbol{X}}(\bullet) \quad \text{and} \quad \operatorname{Div}(\bullet) = \nabla_{\boldsymbol{X}} \cdot (\bullet), \tag{4.2}$$

and with respect to the current configuration

$$\operatorname{grad}(\bullet) = \nabla_{\boldsymbol{x}}(\bullet) \quad \text{and} \quad \operatorname{div}(\bullet) = \nabla_{\boldsymbol{x}} \cdot (\bullet). \tag{4.3}$$

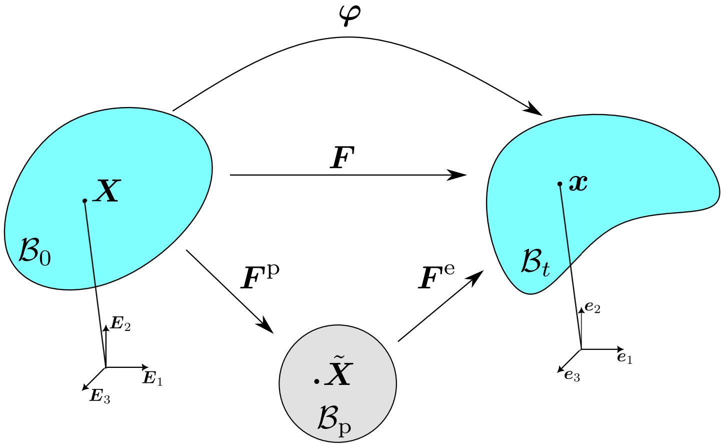


Figure 4.1: Classic nonlinear kinematics with the plastic intermediate configuration.

The deformation gradient can be expressed as

$$\boldsymbol{F} := \mathrm{Grad}(\boldsymbol{\varphi}) = \nabla_{\boldsymbol{X}}\boldsymbol{\varphi} = \frac{\partial\left(\boldsymbol{X}+\boldsymbol{u}\right)}{\partial\boldsymbol{X}} = \nabla_{\boldsymbol{X}}\boldsymbol{x}, \tag{4.4}$$

with $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ as the displacement vector field. Note that, as commonly done in engineering literature, in (4.4) no specific distinction is made between the gradient of the function $\boldsymbol{\varphi}$ and its value $\boldsymbol{x}$ for the derivation of the deformation gradient $\boldsymbol{F}$. This deformation gradient maps an infinitesimal referential line element $\mathrm{d}\boldsymbol{X}$ to its current counterpart $\mathrm{d}\boldsymbol{x}$. The corresponding mapping of volumes, i.e. referential volume $\mathrm{d}V$ to its current counterpart $\mathrm{d}v$, is captured by the Jacobian map J, i.e.

$$J = \det(\boldsymbol{F}) = \frac{\mathrm{d}v}{\mathrm{d}V}. \tag{4.5}$$

Every non-singular second order tensor, such as the deformation gradient, can be decomposed multiplicatively into an orthogonal rotation tensor $\boldsymbol{R} \in \mathcal{SO}(3)$ and a stretch tensor, i.e.

$$\boldsymbol{F} = \boldsymbol{R}\cdot\boldsymbol{U} = \boldsymbol{v}\cdot\boldsymbol{R}, \quad \text{with} \quad \boldsymbol{U} = \boldsymbol{U}^t \text{ and } \boldsymbol{v} = \boldsymbol{v}^t. \tag{4.6}$$

Here, $\boldsymbol{U}$ is the right and $\boldsymbol{v}$ the left stretch tensor. The first split initially stretches a body and then applies a rotation, while the second rotates first and applies the stretch afterwards. Both tensors can be decomposed in their eigenvectors and eigenvalues, called a spectral decomposition, such that

$$\boldsymbol{U} = \sum_{i=1}^{3} \lambda_i \, \boldsymbol{N}_i \otimes \boldsymbol{N}_i, \tag{4.7}$$

$$\boldsymbol{v} = \sum_{i=1}^{3} \tilde{\lambda}_i \, \boldsymbol{n}_i \otimes \boldsymbol{n}_i, \tag{4.8}$$

$$\boldsymbol{R} = \sum_{i=1}^{3} \boldsymbol{n}_i \otimes \boldsymbol{N}_i. \tag{4.9}$$

Based on this, the right and left Cauchy-Green deformation tensors can be defined, i.e.

$$\boldsymbol{C} = \boldsymbol{U}^2 = \boldsymbol{F}^t \cdot \boldsymbol{F} \quad \text{and} \quad \boldsymbol{b} = \boldsymbol{v}^2 = \boldsymbol{F} \cdot \boldsymbol{F}^t \tag{4.10}$$

and their respective spectral decomposition

$$\boldsymbol{C} = \boldsymbol{U}^2 = \sum_{i=1}^{3} \lambda_i^2 \, \boldsymbol{N}_i \otimes \boldsymbol{N}_i \quad \text{and} \quad \boldsymbol{b} = \boldsymbol{v}^2 = \sum_{i=1}^{3} \tilde{\lambda}_i^2 \, \boldsymbol{n}_i \otimes \boldsymbol{n}_i. \tag{4.11}$$

Since the eigenvalues of the deformation gradient and its transpose are equal, the eigenvalues of the deformation tensors have to be equal as well, i.e. $\lambda_i = \tilde{\lambda}_i$. This allows the definition of the Lagrangian strain tensors, defined by [112, 200], i.e.

$$\boldsymbol{E}^m = \begin{cases} \frac{1}{m} \left[\boldsymbol{U}^m - \boldsymbol{I} \right] & m \neq 0 \\ \ln \left[\boldsymbol{U} \right] & m = 0, \end{cases} \tag{4.12}$$

with the identity tensor $\boldsymbol{I}$. The same concept can be applied to the spatial, Eulerian counterpart, i.e.

$$\boldsymbol{e}^m = \begin{cases} \frac{1}{m} \left[\boldsymbol{I} - \boldsymbol{v}^{-m} \right] & m \neq 0 \\ \ln \left[\boldsymbol{v} \right] & m = 0. \end{cases} \tag{4.13}$$

Of special interest in this work are the spatial logarithmic strains, wherein the spectral decomposition reads

$$\boldsymbol{\varepsilon} = \sum_{i=1}^{3} \ln(\lambda_i) \, \boldsymbol{n}_i \otimes \boldsymbol{n}_i. \tag{4.14}$$

Assuming the considered material shows time-dependent behaviour, derivatives w.r.t. time have to be considered. Time derivatives are indicated by a number of dots above the symbol, dependent on the order of derivative. Applied to the displacement field, this yields

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{u}} = \frac{\partial \boldsymbol{x}}{\partial t} \quad \text{and} \quad \ddot{\boldsymbol{x}} = \ddot{\boldsymbol{u}} = \frac{\partial^2 \boldsymbol{x}}{\partial t^2}, \tag{4.15}$$

i.e. the velocity and acceleration, respectively. This allows the definition of the referential and spatial velocity gradient, i.e.

$$\mathrm{Grad}(\dot{\boldsymbol{x}}) = \dot{\boldsymbol{F}} \quad \text{and} \quad \mathrm{grad}(\dot{\boldsymbol{x}}) = \nabla_{\boldsymbol{x}}\dot{\varphi} = \dot{\boldsymbol{F}} \cdot \boldsymbol{F}^{-1} =: \boldsymbol{l}. \tag{4.16}$$

The latter can be additively split into a symmetric deformation rate $\boldsymbol{d}$ and a skew symmetric spin $\boldsymbol{w}$, such that

$$\boldsymbol{d} = \frac{1}{2}\left[\boldsymbol{l} + \boldsymbol{l}^t\right] \quad \text{and} \quad \boldsymbol{w} = \frac{1}{2}\left[\boldsymbol{l} - \boldsymbol{l}^t\right]. \tag{4.17}$$

In addition to the reference and current configuration in Fig. 4.1, a theoretical plastic configuration is considered, cf. [71, 206]. Its introduction is very beneficial for modelling of elasto-plastic materials, as it allows the multiplicative split of the deformation gradient into an elastic $\boldsymbol{F}^{\mathrm{e}}$ and a plastic $\boldsymbol{F}^{\mathrm{p}}$ contribution, i.e.

$$\boldsymbol{F} = \boldsymbol{F}^{\mathrm{e}} \cdot \boldsymbol{F}^{\mathrm{p}}. \tag{4.18}$$

Application of this multiplicative split to the deformation tensors (4.10), yields

$$\boldsymbol{C}^{\mathrm{p}} = [\boldsymbol{F}^{\mathrm{p}}]^t \cdot \boldsymbol{F}^{\mathrm{p}} \quad \text{and} \quad \boldsymbol{b}^{\mathrm{e}} = \boldsymbol{F}^{\mathrm{e}} \cdot [\boldsymbol{F}^{\mathrm{e}}]^t. \tag{4.19}$$

Together with (4.18), this leads to the conversion

$$\boldsymbol{b}^{\mathrm{e}} = \boldsymbol{F} \cdot \boldsymbol{C}^{\mathrm{p}-1} \cdot \boldsymbol{F}^t \quad \Leftrightarrow \quad \boldsymbol{C}^{\mathrm{p}-1} = \boldsymbol{F}^{-1} \cdot \boldsymbol{b}^{\mathrm{e}} \cdot \boldsymbol{F}^{-t}. \tag{4.20}$$

Furthermore, combining the multiplicative split of the deformation gradient (4.18) with the definition of the velocity gradient, yields the additive split

$$\boldsymbol{l} = \dot{\boldsymbol{F}}^{\mathrm{e}} \cdot \dot{\boldsymbol{F}}^{\mathrm{e}-1} + \boldsymbol{F}^{\mathrm{e}} \cdot \dot{\boldsymbol{F}}^{\mathrm{p}} \cdot \boldsymbol{F}^{-1} =: \boldsymbol{l}^{\mathrm{e}} + \boldsymbol{l}^{\mathrm{p}}, \tag{4.21}$$

with the spatial elastic $\boldsymbol{l}^{\mathrm{e}}$ and spatial plastic velocity gradient $\boldsymbol{l}^{\mathrm{p}}$.

4.3 Balance laws

In this section, the balance laws as they find application in modern physics are briefly summarised. More details can be found in e.g. [210, 238].

4.3.1 Balance of mass

In this work, only problems with no mass exchange are considered, i.e. no mass is added or subtracted to the configurations $\dot{M} = \dot{m} = 0$. In this setting, the mass of a body in the referential or current configuration can be computed easily by integrating the specific densities over their respective volume, such that

$$M = \int_{\mathcal{B}_0} \rho_0 \, \mathrm{d}V = \int_{\mathcal{B}_t} \rho \, \mathrm{d}v = m. \tag{4.22}$$

Together with (4.5), the proportion of referential density and current density is equal to the Jacobi map, i.e.

$$\frac{\rho_0}{\rho} = J. \tag{4.23}$$

Combining both of the above equations, allows deriving the rate form of mass continuity in spatial form, such that

$$\dot{M} = 0 = \dot{\rho}_0 = \dot{\overline{\rho\, J}} \quad \Leftrightarrow \quad \dot{\rho}\, J + \rho\, J \operatorname{div}(\dot{\boldsymbol{x}}) = 0. \tag{4.24}$$

4.3.2 Balance of linear and angular momentum

Let $\boldsymbol{L}$ be the linear momentum acting on the body under consideration, defined as

$$\boldsymbol{L} = \int_{\mathcal{B}_0} \rho_0\, \dot{\boldsymbol{x}}\ \mathrm{d}V = \int_{\mathcal{B}_t} \rho\, \dot{\boldsymbol{x}}\ \mathrm{d}v. \tag{4.25}$$

Using the information from Sect. 4.3.1, its time derivative can be expressed as

$$\dot{\boldsymbol{L}} = \int_{\mathcal{B}_t} \rho\, \ddot{\boldsymbol{x}}\, \mathrm{d}v. \tag{4.26}$$

According to Newton's second law of motion, the change of momentum of a body is equal to the sum of all forces acting on it. This statement can thereby by expressed by

$$\dot{\boldsymbol{L}} = \boldsymbol{F}^{\mathrm{vol}} + \boldsymbol{F}^{\mathrm{ext}}, \tag{4.27}$$

where $\boldsymbol{F}^{\mathrm{vol}}$ are the forces within the volume of the body and $\boldsymbol{F}^{\mathrm{ext}}$ external forces applied to the surface of the body. These can be represented in integral form according to

$$\boldsymbol{F}^{\mathrm{vol}} = \int_{\mathcal{B}_t} \rho\, \boldsymbol{b}\, \mathrm{d}v \quad \text{and} \quad \boldsymbol{F}^{\mathrm{ext}} = \int_{\partial\mathcal{B}_t} \boldsymbol{t}\, \mathrm{d}a. \tag{4.28}$$

Therein, $\boldsymbol{b}$ are spatial body forces acting in the body and $\boldsymbol{t}$ spatial surface traction forces. This finally combines into

$$\int_{\mathcal{B}_t} \rho\, \ddot{\boldsymbol{x}}\, \mathrm{d}v = \int_{\mathcal{B}_t} \rho\, \boldsymbol{b}\, \mathrm{d}v + \int_{\partial\mathcal{B}_t} \boldsymbol{t}\, \mathrm{d}a. \tag{4.29}$$

Applying Cauchy's law $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}$ allows reformulating the above equation, where $\boldsymbol{\sigma}$ are the Cauchy stresses and $\boldsymbol{n}$ their outward facing normal vector. This law states that pointwise the forces acting on a surface are equal to the stress tensor of second order times the normal direction of the considers surface. Besides the Cauchy stresses, many additional stress definitions exist. The most important for this work is the first Piola-Kirchoff stress tensor $\boldsymbol{P}^{\mathrm{K}}$, i.e.

$$\boldsymbol{P}^{\mathrm{K}} = J\, \boldsymbol{\sigma} \cdot \boldsymbol{F}^{-t}. \tag{4.30}$$

The first Piola-Kirchoff stress tensor has basis in the current and referential configuration, it measures the stresses in the current configuration according to an infinitesimal volume element in the reference configuration.

Another measurement important when plasticity is modelled are the so-called Mandel stresses $\boldsymbol{\Sigma}$. These are defined within the plastic configuration and are generally used since they enable the use of invariants to define plastic properties. In addition, the spatial Mandel stresses $\boldsymbol{m}$ are utilised within this work. They are be defined with a push-forward of the Mandel stresses $\boldsymbol{\Sigma}$. Both stress measurements read

$$\boldsymbol{\Sigma} = \boldsymbol{F}^{\mathrm{e}} \cdot \frac{\partial \Psi}{\partial \boldsymbol{F}^{\mathrm{e}}} \quad \text{and} \quad \boldsymbol{m} = [\boldsymbol{F}^{\mathrm{e}}]^{-t} \cdot \boldsymbol{\Sigma} \cdot [\boldsymbol{F}^{\mathrm{e}}]^{t} \,. \tag{4.31}$$

Additionally, combining (4.30) and (4.31), allows definition of the relation

$$\boldsymbol{P}^{\mathrm{K}} = \boldsymbol{m} \cdot \boldsymbol{F}^{-t}, \tag{4.32}$$

for the First-Piola Kirchhoff stress tensor and the spatial Mandel stresses, which will find application within the material model presented in this thesis.

Inserting Cauchy's law into (4.29) and application of the divergence theorem finally results in the weak form of linear momentum

$$\int_{\mathcal{B}_t} \rho \, \ddot{\boldsymbol{x}} \, \mathrm{d}v = \int_{\mathcal{B}_t} \rho \, \boldsymbol{b} \, \mathrm{d}v + \int_{\mathcal{B}_t} \boldsymbol{\sigma} \cdot \boldsymbol{n} \, \mathrm{d}v. \tag{4.33}$$

In this thesis however, only quasi-static cases are considered. Therefore, the acceleration is assumed to be zero, i.e. $\ddot{\boldsymbol{x}} = \boldsymbol{0}$, which leads to the equilibrium condition which only considering internal and external forces.

Similarly to the balance of linear momentum, the balance of angular momentum states that the change of angular momentum of a body w.r.t. a fixed point $\boldsymbol{x}_0$, is equal to the sum of all moments exerted on the body, due to external volume of surfaces forces with respect to the same fixed point. The necessary conclusion of this regarding this work is the requirement of a symmetric Cauchy stress tensor, i.e. $\boldsymbol{\sigma} = \boldsymbol{\sigma}^t$.

4.3.3 First law of thermodynamics

The first law of thermodynamics postulates the conversation of energy, that is one type of energy can only be converted into another kind of energy. In the field of continuum mechanics, the total energy of the system E consists of the internal energy U and the kinetic energy K. A change in energy over time for a considered system can only occur due to external influence. External sources consist of external applied power P and heat supply Q. This yields

$$\dot{E} = P + Q. \tag{4.34}$$

The time derivatives of kinetic and internal part are specified as

$$\dot{K} = \int_{\mathcal{B}_t} \rho \, \ddot{\boldsymbol{x}} \cdot \dot{\boldsymbol{x}} \, \mathrm{d}v \quad \text{and} \quad \dot{U} = \int_{\mathcal{B}_t} \rho \, \dot{u} \, \mathrm{d}v. \tag{4.35}$$

In the field of continuum mechanics, the internal energy contribution u is generally specified in terms of Helmholtz free energy Ψ, temperature Θ and entropy s, such that

$$u = \Psi + \Theta\, s \quad \Rightarrow \quad \dot{u} = \dot{\Psi} + \dot{\Theta}\, s + \Theta\, \dot{s}. \tag{4.36}$$

The external contribution includes the heat flux $\boldsymbol{q}$ which is applied through the surface of the body and internal heat source r, such that

$$P = \int_{\partial \mathcal{B}_t} \boldsymbol{t} \cdot \dot{\boldsymbol{x}} \,\mathrm{d}a + \int_{\mathcal{B}_t} \rho\, \boldsymbol{b} \cdot \dot{\boldsymbol{x}} \,\mathrm{d}v \quad \text{and} \quad Q = -\int_{\partial \mathcal{B}_t} \boldsymbol{q} \cdot \boldsymbol{n} \,\mathrm{d}a + \int_{\mathcal{B}_t} \rho\, r \,\mathrm{d}v. \tag{4.37}$$

Combination of all quantities and application of the divergence theorem together with balance of linear momentum finally leads to the local form

$$\dot{\Psi} + \dot{\Theta}\, s + \Theta\, \dot{s} = \boldsymbol{\sigma} : \boldsymbol{d} + \rho\, r - \operatorname{div}(\boldsymbol{q}), \tag{4.38}$$

wherein $\boldsymbol{\sigma} : \boldsymbol{d}$ is the so-called specific stress power.

Remark 4.1 Note, that here s denotes the entropy. This is not to be confused with the quantity $\boldsymbol{s}$ which is used describe the design variables of an optimisation.

4.3.4 Second law of thermodynamics

The second law considers the direction of energy flow and the inherent irreversibility of entropy. It postulates that the entropy rate $\dot{S}$ is always greater or equal than the external entropy supply R^{ext}, i.e. $\dot{S} - R^{\text{ext}} \geq 0$. This results in

$$\int_{\mathcal{B}_t} \rho\, \dot{s} \,\mathrm{d}v + \int_{\partial \mathcal{B}_t} \frac{1}{\Theta} \boldsymbol{q} \cdot \boldsymbol{n} \,\mathrm{d}a - \int_{\mathcal{B}_t} \frac{\rho\, r}{\Theta} \,\mathrm{d}v \geq 0. \tag{4.39}$$

Combining this definition with the first law of thermodynamics (4.38), as well as restructuring the equation, leads to the local dissipation inequality, also known as the Clausius-Duhem inequality

$$\mathcal{D} = \boldsymbol{\sigma} : \boldsymbol{d} - \rho \left[\dot{\Psi} + s\, \dot{\Theta} \right] - \frac{1}{\Theta}\, \boldsymbol{q} \cdot \operatorname{grad}(\Theta) \geq 0. \tag{4.40}$$

In the context of this work, thermal contributions are neglected and an isothermal process, $\Theta = \text{const.}$, is assumed. This reduces the inequality to

$$\mathcal{D} = \boldsymbol{\sigma} : \boldsymbol{d} - \rho \dot{\Psi} \geq 0. \tag{4.41}$$

4.4 Nonlocal damage regularisation

The main problem with classical, local damage models, from a mathematical view point, is the loss of ellipticity of the governing equations, i.e. the mechanical tangent is no longer positive definite [144]. From a numerics viewpoint, this leads to an inherent mesh-dependency in the simulation results. That is, the accuracy of the finite element (FE) simulation does not increase with higher mesh density [25, 181]. Instead, the local damage may start to predict damage behaviour to occur in smaller areas of the FE mesh, leading to mesh-dependent localisation. Due to this background, multiple possible approaches to solve this problem exist.

Fracture energy concept. One approach to circumvent this problem is by coupling the numerics, i.e. FE discretisation, with physical data in form of the fracture energy, cf. [23]. This idea does not solve the underlying mathematical problems of the equations which lead to the mesh-dependency, but instead adapt the results to be in accordance to physics. A specific fracture energy is introduced and combination with mesh properties adjusted, such that the physical fracture energy is reproduced. The main benefit of this approach is the quick calibration for a given problem. However, in the similar vein, it is only appropriate for the calibrated mesh-size and requires recalibration if another FE mesh is considered. While this solves the issues regarding the validity of the computed results, the damage will still localise in one element row, cf. [123].

Viscous regularisation. This approach, cf. [69], aims to resolve the mesh-dependency by introducing a rate-dependent damage evolution. If the chosen time discretisation is appropriate, the resulting mechanics behaviour and damage distribution is mesh-independent. Furthermore, the mechanical tangent remains positive definite. This approach has been expanded and applied in further literature for many different fields of application, cf. [5, 31, 140]. While this approach resolves the dependency of the damage evolution on the numerical description of geometry, it shifts this dependency into the time space. Therefore, the results now depend on the chosen time discretisation. If the damage behaviour in this approach is coupled to other rate-dependent mechanisms, e.g. temperature effects, interference between the descriptions might occur.

Nonlocal models. Another approach to overcome the mesh-dependency in continuum damage modelling is by means of nonlocal models. These models introduce nonlocal damage quantities based on certain concepts, which therefore no longer limits damage to their respective finite element and take surrounding information into consideration. In [22, 24], a nonlocal damage variable is defined as an integral value of a pointwise defined damage quantity. Another, more common, method in nonlocal damage modelling is the utilisation of gradient information. In this setting, the gradient of the nonlocal damage variable is introduced as the additional information within the underlying model description. This concept results from a Taylor-series expansion, excluding information of order two and higher.

4.4.1 Micromorphic approach

While in theory many different approaches are possible, like moving the local damage variable to the global scale and solving the constitutive equations there, cf. [103, 144], nowadays the so-called micromorphic approach is often applied. The basic idea is to introduce a nonlocal damage variable ϕ, which is coupled to a local damage quantity d by means of a penalty approach, cf. [63, 80, 180]. This enables regularisation of the nonlocal variable, while all constitutive equations can be solved on the local scale. The nonlocal energy contribution thus reads

$$\Psi^{\mathrm{nl}} = \frac{c_{\mathrm{d}}}{2} \|\nabla_{\boldsymbol{X}} \phi\|^2 + \frac{\beta_{\mathrm{d}}}{2} [\phi - d]^2 , \tag{4.42}$$

where the first additive term regularises the nonlocal damage and is weighted by the regularisation parameter c_{d}. The second term couples the nonlocal damage ϕ to the local damage quantity d, forcing equilibrium of these two quantities. The above framework utilises the referential gradient $\nabla_{\boldsymbol{X}} \phi$. Alternatively, a spatial gradient $\nabla_{\boldsymbol{x}} \phi$ could also be employed, see Remark 1 of [215]. This nonlocal energy contribution can be coupled to a local part, resulting in the total Helmholtz energy

$$\Psi = \Psi^{\mathrm{loc}}(\boldsymbol{F}, d) + \Psi^{\mathrm{nl}}(\boldsymbol{F}, d, \phi, \nabla_{\boldsymbol{X}} \phi). \tag{4.43}$$

One benefit of this approach, due to the setup of the energy term, is the choice of a local contribution Ψ^{loc} can, in theory, be of any type. Furthermore, this eases the effort for the numerical implementation, since this approach does not require solving any of the resulting Karush-Kuhn-Tucker conditions on the global scale, see [103, 144] where this is required.

4.4.2 Weak form of equilibrium

The total energy for a quasi-static system splits into an internal and external part and reads

$$\begin{aligned} \Pi &= \Pi_{\mathrm{int}} + \Pi_{\mathrm{ext}} = \Pi_{\mathrm{int}}^{\mathrm{loc}} + \Pi_{\mathrm{int}}^{\mathrm{nl}} + \Pi_{\mathrm{ext}} \\ &= \int_{\mathcal{B}_0} \Psi \, \mathrm{d}V - \int_{\mathcal{B}_0} \rho_0 \, \boldsymbol{b}_0 \cdot \boldsymbol{\varphi} \, \mathrm{d}V - \int_{\partial \mathcal{B}_0} \bar{\boldsymbol{t}}_0 \cdot \boldsymbol{\varphi} \, \mathrm{d}A. \end{aligned} \tag{4.44}$$

The external energy contribution includes the body forces $\boldsymbol{b}_0$, which are weighted by the referential mass density ρ_0, and the traction forces $\bar{\boldsymbol{t}}_0$ applied on the surface $\partial \mathcal{B}_0$. The postulate of minimum potential energy, i.e.

$$\delta_{\boldsymbol{\varphi}} \Pi = 0, \tag{4.45}$$

$$\delta_{\phi} \Pi = 0, \tag{4.46}$$

results in the two weak forms for the mechanical and damage part

$$r^{\varphi} = \int_{\mathcal{B}_0} \frac{\partial \Psi}{\partial \boldsymbol{F}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \, \mathrm{d}V - \rho_0 \int_{\mathcal{B}_0} \boldsymbol{b}_0 \cdot \boldsymbol{\eta}^{\varphi} \, \mathrm{d}V - \int_{\partial \mathcal{B}_0} \bar{\boldsymbol{t}}_0 \cdot \boldsymbol{\eta}^{\varphi} \, \mathrm{d}A = 0, \tag{4.47}$$

$$r^{\phi} = \int_{\mathcal{B}_0} \left[\frac{\partial \Psi}{\partial \nabla_{\boldsymbol{X}} \phi} \cdot \nabla_{\boldsymbol{X}} \eta^{\phi} + \frac{\partial \Psi}{\partial \phi} \eta^{\phi} \right] \mathrm{d}V = 0, \tag{4.48}$$

and can be combined into

$$\boldsymbol{r} = \begin{bmatrix} r^{\varphi} \\ r^{\phi} \end{bmatrix} = \boldsymbol{0}. \tag{4.49}$$

To differentiate between the variations in the above equations and in the upcoming ones in Chap. 5, the test functions $\boldsymbol{\eta}^{\varphi}$ and η^{ϕ}, using (2.10), are introduced here. These weak forms are subsequently discretised in Sect. 4.6.1 for the FE implementation. The partial derivatives in the above equations can be identified as the first Piola-Kirchhoff stress tensor $\boldsymbol{P}^{\mathrm{K}}$, as well as the nonlocal damage driving forces $\boldsymbol{Y}$ and Y, i.e.

$$\boldsymbol{P}^{\mathrm{K}} := \frac{\partial \Psi}{\partial \boldsymbol{F}}, \qquad \boldsymbol{Y} := \frac{\partial \Psi}{\partial \nabla_{\boldsymbol{X}} \phi}, \qquad Y := \frac{\partial \Psi}{\partial \phi}. \tag{4.50}$$

4.5 Constitutive model

The local constitutive model utilises the multiplicative split presented in Sect. 4.2. Combining (4.20) and (4.11), spectral decomposition of the elastic left Cauchy-Green deformation tensor reads

$$\boldsymbol{b}^{\mathrm{e}} = \boldsymbol{F}^{\mathrm{e}} \cdot [\boldsymbol{F}^{\mathrm{e}}]^{t} = \sum_{i=1}^{3} [\lambda_i^{\mathrm{e}}]^2 \, \boldsymbol{n}_i \otimes \boldsymbol{n}_i. \tag{4.51}$$

Using the elastic eigenvalues of this decomposition, and combination with (4.14), yields the elastic logarithmic strains

$$\varepsilon^{\mathrm{e}} := \sum_{i=1}^{3} \ln(\lambda_i^{\mathrm{e}}) \, \boldsymbol{n}_i \otimes \boldsymbol{n}_i = \sum_{i=1}^{3} \varepsilon_i^{\mathrm{e}} \, \boldsymbol{n}_i \otimes \boldsymbol{n}_i, \tag{4.52}$$

which can be additionally split into a volumetric and an isochoric part, i.e.

$$\varepsilon^{\mathrm{e,vol}} := \sum_{i=1}^{3} \varepsilon_i^{\mathrm{e}}, \quad \boldsymbol{\varepsilon}^{\mathrm{e,iso}} := \sum_{i=1}^{3} \left[\varepsilon_i^{\mathrm{e}} - \frac{1}{3} \varepsilon^{\mathrm{e,vol}} \right] \boldsymbol{n}_i \otimes \boldsymbol{n}_i. \tag{4.53}$$

With this split in strains, the local Helmholtz energy Ψ^{loc} is also split into a volumetric part Ψ^{vol} and an isochoric part Ψ^{iso}. With the additional plastic contribution Ψ^{p}, the local Helmholtz energy is specified as

$$\begin{aligned} \Psi^{\mathrm{loc}}(\boldsymbol{\varepsilon}^{\mathrm{e}},d,\alpha) &= \Psi^{\mathrm{vol}}(\boldsymbol{\varepsilon}^{\mathrm{e}},d) + \Psi^{\mathrm{iso}}(\boldsymbol{\varepsilon}^{\mathrm{e}},d) + \Psi^{\mathrm{p}}(\alpha) \\ &= \frac{K}{2} f^{\mathrm{vol}}(d) \, [\mathrm{tr}(\boldsymbol{\varepsilon}^{\mathrm{e}})]^2 + G \, f^{\mathrm{iso}}(d) \, \boldsymbol{\varepsilon}^{\mathrm{e,iso}} : \boldsymbol{\varepsilon}^{\mathrm{e,iso}} \; + \frac{h}{n_{\mathrm{p}}+1} \, \alpha^{n_{\mathrm{p}}+1}, \end{aligned} \tag{4.54}$$

where K denotes the compression and G the shear modulus. Additionally, the plastic hardening modulus h and the hardening exponent n_{p} are introduced and together with the internal plastic variable α form the plastic energy contribution. Both elastic terms additional introduce the damage functions $f^{\bullet}$. These replace the typical $[1-d]$ approach often found in other models in literature, and are defined as

$$f^{\bullet}(d) := \exp(-\eta \, \xi_{\bullet} d) \tag{4.55}$$

with the internal damage variable $d \in [0,1[$, such that

$$f^\bullet : \left\{ \mathbb{R}_0^+ \to]0,1] \,\middle|\, f^\bullet(0) = 1, \lim_{d \to \infty} f^\bullet(d) = 0 \right\}. \tag{4.56}$$

The parameter η influences the rate of degradation in general, which can be further specified with $\xi_\bullet$ for each respective contribution within the model, e.g. volumetric or isochoric contribution within (4.54). This approach, in contrast to the $[1-d]$ definition, does not limit the damage variable to an upper bound. The local damage quantity d is only defined to be non-negative and its increase throughout loading is unrestricted. This however, has a slight drawback regarding immediate translation of damage variable to damage state. Since this quantity no longer correlates $d = 0 \to$ no damage and $d = 1 \to$ failure, the specific damage function or their parameters need to be explicitly known to analyse a resulting damage evolution. The main benefit of this approach, however, is in its numerical implementation and the algorithmic framework of the model.

For simplicity, in this work the notation

$$[f^\bullet]'(d) = \frac{\partial f^\bullet}{\partial d} \tag{4.57}$$

is additionally introduced here.

The dissipation inequality, following (4.41) and using (4.16), reads

$$\mathcal{D} = \boldsymbol{m} : \boldsymbol{l} - \dot{\varPsi} \geq 0. \tag{4.58}$$

Solving the dissipation inequality by applying the postulate of minimum dissipation, as well as the procedure proposed by Coleman and Noll, cf. [54], the spatial Mandel stresses, isotropic hardening stress and the local damage driving force can be specified, i.e.

$$\boldsymbol{m} := 2\frac{\partial \varPsi}{\partial \boldsymbol{b}^\mathrm{e}} \cdot \boldsymbol{b}^\mathrm{e}, \quad \beta := -\frac{\partial \varPsi}{\partial \alpha}, \quad q := -\frac{\partial \varPsi}{\partial d}, \tag{4.59}$$

and, using (4.21), leads to the reduced dissipation inequality

$$\mathcal{D}^\mathrm{red} = \boldsymbol{m} : \boldsymbol{l}^\mathrm{p} + \beta\,\dot{\alpha} + q\,\dot{d} \geq 0. \tag{4.60}$$

To govern the evolution of plasticity and damage, respectively, two separate dissipation potential $\varPhi^\mathrm{p}$ and $\varPhi^\mathrm{d}$ are defined

$$\varPhi^\mathrm{p}(\boldsymbol{m},\beta,d) := \|\mathrm{dev}(\boldsymbol{m}_\mathrm{eff})\| - \sqrt{\frac{2}{3}}\left[\sigma_{\mathrm{y}0} - \beta\right], \tag{4.61}$$

$$\varPhi^\mathrm{d}(q,d,\alpha) := q_\mathrm{eff} - q_\mathrm{min}\left[1 - f^q(d)\right]^{n_\mathrm{d}}, \tag{4.62}$$

where $\mathrm{dev}(\bullet) = \bullet - \frac{1}{3}\mathrm{tr}(\bullet) : \boldsymbol{I}$ denotes the deviatoric part of a stress measurement. The dissipation potential utilises effective driving forces, which introduce strong coupling between plastic and damage evolution and are defined as

$$\boldsymbol{m}_{\text{eff}} := \boldsymbol{m}\,\hat{f}^{\boldsymbol{m}}(d), \tag{4.63}$$

$$q_{\text{eff}} := q\,\hat{f}^{\alpha}(\alpha), \tag{4.64}$$

where modified coupling functions are introduced as

$$\hat{f}^{\boldsymbol{m}}(d) = \frac{1+\epsilon}{f^{\boldsymbol{m}}(d)+\epsilon} \quad \text{and} \quad \hat{f}^{\alpha}(\alpha) = \frac{1+\epsilon}{f^{\alpha}(\alpha)+\epsilon}, \tag{4.65}$$

with the perturbation value set to $\epsilon = 1 \times 10^{-3}$.

The associative evolution equations are additionally derived as

$$\begin{aligned} \boldsymbol{l}^{\text{p}} &= \lambda^{\text{p}} \frac{\partial \varPhi^{\text{p}}}{\partial \boldsymbol{m}} = \lambda^{\text{p}} \hat{f}^{\boldsymbol{m}} \boldsymbol{\nu}, \\ \dot{\alpha} &= \lambda^{\text{p}} \frac{\partial \varPhi^{\text{p}}}{\partial \beta} = \lambda^{\text{p}} \sqrt{\frac{2}{3}}, \\ \dot{d} &= \lambda^{\text{d}} \frac{\partial \varPhi^{\text{d}}}{\partial q} = \lambda^{\text{d}} \hat{f}^{\alpha}, \end{aligned} \tag{4.66}$$

where the two Lagrange parameters λ^{p} and λ^{d} are introduced. Together with their respective dissipation potentials, they combine into the two sets of Karush-Kuhn-Tucker conditions

$$\varPhi^{\text{p}} \quad \leq 0, \quad \lambda^{\text{p}} \geq 0, \quad \lambda^{\text{p}} \varPhi^{\text{p}} = 0, \tag{4.67}$$

$$\varPhi^{\text{d}} \quad \leq 0, \quad \lambda^{\text{d}} \geq 0, \quad \lambda^{\text{d}} \varPhi^{\text{d}} = 0, \tag{4.68}$$

which, as a physical interpretation, govern initiation and termination of plastic and damage evolution under loading or unloading, respectively. The plastic dissipation potential is of classic von Mises type with initial yield stress σ_{y0}. The damage onset in the damage dissipation potential is governed by a threshold q_{min}. This quantity is multiplied by the term in the bracket, which decreases in value with increasing damage accumulation. With the parameter $n_{\text{d}} < 1$, this results in higher damage thresholds as damage increases. The aim of this formulation is to allow for a smoother damage evolution, as well as a more ductile response of the material, cf. [215]. Of note here, for $n_{\text{d}} = 0$ the term in the bracket is disabled, and damage initiates only after the initial, constant threshold value q_{min} is surpassed.

4.6 Numerical implementation

For the numerical implementation and computation thereof, the field variables and hence the weak forms and equations of the constitutive model have to be discretised. The former is specified, wherein the same discretisation will be again applied in Sect. 5. The algorithmic treatment of the latter is thoroughly explained and the implementations sketched. The discrete variables are now denoted, according to Sect. 2.1, as bold faced

Roman letters. For example, the continuous $\boldsymbol{X}$ is now replaced with its discrete counterpart $\mathbf{X}$. Some of the required equations and their derivations, however, are omitted in this section for readability purposes. These are listed in Appendix A.1 -A.3, wherein the latter briefly summarises the $\bar{\boldsymbol{F}}$-method utilised. Finally, to verify the correct implementation of the model, simulation results from the implemented model were compared to data from the original source [215]. These can be found in Appendix A.4.

4.6.1 Finite element method

For the numerical implementation, a FE discretisation is applied to the weak forms in (4.47) and (4.48). The domain $\mathcal{B}_0$ is discretised by n_{el} finite elements and n_{np} nodes, i.e.

$$\mathcal{B}_0 \approx \mathcal{B}_0^h = \bigcup_{e=1}^{n_{\mathrm{el}}} \mathcal{B}_0^e. \tag{4.69}$$

Each finite element e is associated with $n_{\mathrm{en}}^{\varphi}$ nodes with three displacement degrees of freedom $\boldsymbol{\varphi}$, and n_{en}^{ϕ} nodes with one nonlocal damage degree of freedom ϕ. The referential geometry description $\boldsymbol{X}$ is interpolated with shape functions N_C^X, i.e.

$$\boldsymbol{X} \approx \boldsymbol{X}^h = \sum_{C=1}^{n_{\mathrm{en}}^{\varphi}} \mathbf{X}_C N_C^X. \tag{4.70}$$

Application of an isoparametric concept, i.e. $N^X = N^{\varphi} = N^{\phi}$, and the Bubnov-Galerkin approach, this results in the approximations

$$\begin{aligned}
\boldsymbol{\varphi} \approx \boldsymbol{\varphi}^h &= \sum_{A=1}^{n_{\mathrm{en}}^{\varphi}} \boldsymbol{\upvarphi}_A N_A^{\varphi}, & \boldsymbol{\eta}^{\varphi} \approx \boldsymbol{\eta}^{\varphi h} &= \sum_{A=1}^{n_{\mathrm{en}}^{\varphi}} \boldsymbol{\upeta}_A^{\varphi} N_A^{\varphi}, \\
\phi \approx \phi^h &= \sum_{B=1}^{n_{\mathrm{en}}^{\phi}} \upphi_B N_B^{\phi}, & \eta^{\phi} \approx \eta^{\phi h} &= \sum_{B=1}^{n_{\mathrm{en}}^{\phi}} \upeta_B^{\phi} N_B^{\phi}, \\
\delta \boldsymbol{X} \approx \delta \boldsymbol{X}^h &= \sum_{C=1}^{n_{\mathrm{en}}^{\varphi}} \delta \mathbf{X}_C N_C^X,
\end{aligned} \tag{4.71}$$

for the field variables and test functions, and furthermore

$$\begin{aligned}
\nabla_{\boldsymbol{X}} \boldsymbol{\varphi} \approx \nabla_{\boldsymbol{X}} \boldsymbol{\varphi}^h &= \sum_{A=1}^{n_{\mathrm{en}}^{\varphi}} \boldsymbol{\upvarphi}_A \otimes \nabla_{\boldsymbol{X}} N_A^{\varphi}, & \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \approx \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi h} &= \sum_{A=1}^{n_{\mathrm{en}}^{\varphi}} \boldsymbol{\upeta}_A^{\varphi} \otimes \nabla_{\boldsymbol{X}} N_A^{\varphi}, \\
\nabla_{\boldsymbol{X}} \phi \approx \nabla_{\boldsymbol{X}} \phi^h &= \sum_{B=1}^{n_{\mathrm{en}}^{\phi}} \upphi_B \nabla_{\boldsymbol{X}} N_B^{\phi}, & \nabla_{\boldsymbol{X}} \eta^{\phi} \approx \nabla_{\boldsymbol{X}} \eta^{\phi^h} &= \sum_{B=1}^{n_{\mathrm{en}}^{\phi}} \upeta_B^{\phi} \nabla_{\boldsymbol{X}} N_B^{\phi}, \\
\nabla_{\boldsymbol{X}} \boldsymbol{X} \approx \nabla_{\boldsymbol{X}} \boldsymbol{X}^h &= \sum_{C=1}^{n_{\mathrm{en}}^{\varphi}} \mathbf{X}_C \otimes \nabla_{\boldsymbol{X}} N_C^X, & \nabla_{\boldsymbol{X}} \delta \boldsymbol{X} \approx \nabla_{\boldsymbol{X}} \delta \boldsymbol{X}^h &= \sum_{C=1}^{n_{\mathrm{en}}^{\varphi}} \delta \mathbf{X}_C \otimes \nabla_{\boldsymbol{X}} N_C^X,
\end{aligned}$$

(4.72)

for their gradients. Insertion of these approximations into the weak forms (4.47) and (4.48) yields

$$\delta_{\boldsymbol{\varphi}} \mathit{\Pi}^h_{eA} = \boldsymbol{\eta}^\varphi \cdot \left[\int_{\mathcal{B}_0^e} \mathbf{P}^{\mathrm{K}} \cdot \nabla_{\boldsymbol{X}} N_A^\varphi \, \mathrm{d}V - \rho_0 \int_{\mathcal{B}_0^e} \mathbf{b}_0 \, N_A^\varphi \, \mathrm{d}V - \int_{\partial\mathcal{B}_0^e} \bar{\mathbf{t}}_0 \, N_A^\varphi \, \mathrm{d}A \right] = 0, \tag{4.73}$$

$$\delta_\phi \mathit{\Pi}^h_{eB} = \eta^\phi \left[\int_{\mathcal{B}_0^e} \left[\mathbf{Y} \cdot \nabla_{\boldsymbol{X}} N_B^\phi + Y N_B^\phi \right] \mathrm{d}V \right] = 0. \tag{4.74}$$

The first integral in (4.73) can be identified as the internal forces with the latter two representing the external contribution. (4.73) and (4.74) have to be fulfilled for arbitrary values of the test functions $\boldsymbol{\eta}^\varphi$ and η^ϕ, which subsequently allows the above equations to be rewritten as the sum of forces, i.e.

$$\mathbf{r}_e = \begin{bmatrix} \mathbf{r}^\varphi_{eA} \\ \mathrm{r}^\phi_{eB} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{\mathrm{int},\varphi}_{eA} - \mathbf{f}^{\mathrm{ext},\varphi}_{eA} \\ \mathrm{f}^{\mathrm{int},\phi}_{eB} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}. \tag{4.75}$$

Assembly of the elemental quantities into a global FE residual consequently leads to

$$\mathbf{r}^\varphi = \mathop{\mathsf{A}}_{e}^{n_{\mathrm{el}}} \mathbf{r}^\varphi_{eA}, \quad \mathbf{r}^\phi = \mathop{\mathsf{A}}_{e}^{n_{\mathrm{el}}} \mathrm{r}^\phi_{eB}, \quad \mathbf{r} = \mathop{\mathsf{A}}_{e}^{n_{\mathrm{el}}} \mathbf{r}_e. \tag{4.76}$$

While quasi-static problems are analysed here, which do not explicitly depend on the time discretisation, the loads are applied by pseudo-time increments, i.e. $t_{n+1} = t_n + \mathit{\Delta} t$. To solve the nonlinear system of equations (4.75) at each respective time step t_{n+1}, the Newton-Raphson method is utilised. Therefore, the system is linearised by a Taylor series expansion, i.e.

$$\mathbf{r}^\varphi_{n+1} = \mathbf{r}^\varphi_n + \mathit{\Delta}\mathbf{r}^\varphi_n = \mathbf{0}, \tag{4.77}$$

$$\mathbf{r}^\phi_{n+1} = \mathbf{r}^\phi_n + \mathit{\Delta}\mathbf{r}^\phi_n = \mathbf{0}, \tag{4.78}$$

where terms of higher order are neglected. The respective nodal increments result in

$$\mathit{\Delta}\mathbf{r}^\varphi_n = \sum_{C=1}^{n^\varphi_{\mathrm{en}}} \frac{\mathrm{d}\mathbf{r}^\varphi_n}{\mathrm{d}\boldsymbol{\varphi}_C} \cdot \mathit{\Delta}\boldsymbol{\varphi}_C + \sum_{D=1}^{n^\phi_{\mathrm{en}}} \frac{\mathrm{d}\mathbf{r}^\varphi_n}{\mathrm{d}\phi_D} \Delta\phi_D, \tag{4.79}$$

$$\mathit{\Delta}\mathbf{r}^\phi_n = \sum_{C=1}^{n^\phi_{\mathrm{en}}} \frac{\mathrm{d}\mathbf{r}^\phi_n}{\mathrm{d}\boldsymbol{\varphi}_C} \cdot \mathit{\Delta}\boldsymbol{\varphi}_C + \sum_{D=1}^{n^\phi_{\mathrm{en}}} \frac{\mathrm{d}\mathbf{r}^\phi_n}{\mathrm{d}\phi_D} \Delta\phi_D, \tag{4.80}$$

with $\mathit{\Delta}\boldsymbol{\varphi}_C = \boldsymbol{\varphi}_{n+1,C} - \boldsymbol{\varphi}_{n,C}$ and $\mathit{\Delta}\phi_D = \phi_{n+1,D} - \phi_{n,D}$. The total derivatives in the above equations represent the stiffness terms that are calculated element-wise and assembled into the global stiffness matrix $\mathbf{K}$

$$\mathbf{K}_e = \begin{bmatrix} \mathbf{K}_e^{\varphi\varphi} \, \mathbf{K}_e^{\varphi\phi} \\ \mathbf{K}_e^{\phi\varphi} \, \mathbf{K}_e^{\phi\phi} \end{bmatrix}, \quad \mathbf{K} = \mathop{\mathsf{A}}_{e}^{n_{\text{el}}} \mathbf{K}_e, \quad \mathbf{K} \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}} \tag{4.81}$$

The stiffness matrix has the dimension $n_{\text{dof}} \times n_{\text{dof}}$, where $n_{\text{dof}} = [n_{\text{dim}}^{\varphi} + 1] \cdot n_{\text{np}}$. Here, n_{dof} are the total number of degrees of freedom, n_{dim}^{φ} the number of dimensions for the deformation field and n_{np} the number of node points. The element-wise stiffness contributions thus read

$$\mathbf{K}_{eAC}^{\varphi\varphi} = \frac{\mathrm{d}\mathbf{r}_{eA,n}^{\varphi}}{\mathrm{d}\boldsymbol{\varphi}_C} = \int_{\mathcal{B}_0^e} [\mathbf{I} \otimes \nabla_{\boldsymbol{X}} N_A^{\varphi}] : \frac{\partial \mathbf{P}^{\text{K}}}{\partial \mathbf{F}} \cdot \nabla_{\boldsymbol{X}} N_C^{\varphi} \, \mathrm{d}V, \tag{4.82}$$

$$\mathbf{K}_{eAD}^{\varphi\phi} = \frac{\mathrm{d}\mathbf{r}_{eA,n}^{\varphi}}{\mathrm{d}\phi_D} = \int_{\mathcal{B}_0^e} [\mathbf{I} \otimes \nabla_{\boldsymbol{X}} N_A^{\varphi}] : \frac{\partial \mathbf{P}^{\text{K}}}{\partial \phi} N_D^{\phi} \, \mathrm{d}V, \tag{4.83}$$

$$\mathbf{K}_{eBC}^{\phi\varphi} = \frac{\mathrm{d}\mathbf{r}_{eB,n}^{\phi}}{\mathrm{d}\boldsymbol{\varphi}_C} = \int_{\mathcal{B}_0^e} N_B^{\phi} \frac{\partial Y}{\partial \mathbf{F}} \cdot \nabla_{\boldsymbol{X}} N_C^{\varphi} \, \mathrm{d}V, \tag{4.84}$$

$$\mathbf{K}_{eBD}^{\phi\phi} = \frac{\mathrm{d}\mathbf{r}_{eB,n}^{\phi}}{\mathrm{d}\phi_D} = \int_{\mathcal{B}_0^e} N_B^{\phi} \frac{\partial Y}{\partial \phi} N_D^{\phi} + \nabla_{\boldsymbol{X}} N_B^{\phi} \cdot \frac{\partial \mathbf{Y}}{\partial \nabla_{\boldsymbol{X}} \phi} \cdot \nabla_{\boldsymbol{X}} N_D^{\phi} \, \mathrm{d}V, \tag{4.85}$$

where $\mathbf{I}$ represents the matrix form of the second order identity tensor. The total derivatives in their respective stiffness contributions will not be listed here but can be found in Appendix A.1.

Finally, assembly of the nodal increments

$$\varDelta\mathbf{w}_e = \begin{bmatrix} \varDelta\boldsymbol{\varphi}_{eC} \\ \varDelta\phi_{eD} \end{bmatrix}, \quad \varDelta\mathbf{w} = \mathop{\mathsf{A}}_{e}^{n_{\text{el}}} \mathbf{w}_e, \tag{4.86}$$

results in the global linearised form

$$\mathbf{r} + \mathbf{K} \cdot \varDelta\mathbf{w} = \mathbf{0}. \tag{4.87}$$

To solve this linear system of equations, the Dirichlet boundary conditions have to be considered. The index a is introduced to denote the unknown degrees of freedom and b for the known, i.e. prescribed, degrees of freedom. Hence, the above system can be reformulated, such that

$$\begin{bmatrix} \mathbf{r}_a \\ \mathbf{r}_b \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix}. \tag{4.88}$$

The resulting submatrix, i.e. reduced tangent stiffness matrix, $\mathbf{K}_{aa}$ is positive definite and therefore allows computation of its inverse. Finally, this enables solving of the global linearised problem for the unknown increment $\varDelta\boldsymbol{w}_a$, such that

$$\varDelta\mathbf{w}_a = \mathbf{K}_{aa}^{-1} \cdot [\mathbf{r}_a + \mathbf{K}_{ab} \cdot \mathbf{w}_b]. \tag{4.89}$$

4.6.2 Algorithmic treatment

The evolution equations are algorithmically solved by means of an active-set strategy. For the algorithmic treatment, a time discretisation is required. The scalar valued evolution equations for the internal plastic variable α and internal damage variable d are solved via an implicit backward-Euler scheme. For the evolution of plastic deformations, an implicit exponential integration scheme, cf. [210], is required to fulfil the plastic incompressibility condition $\det(\mathbf{F}^{\mathrm{p}}) = 1$. With the pseudo-time increments $t_{n+1} = t_n + \Delta t$, this yields

$$\mathbf{F}^{\mathrm{p}} = \exp\left(\Delta\lambda^{\mathrm{p}} \boldsymbol{\nu}_0 \hat{f}^{\boldsymbol{m}}(d)\right) \cdot \mathbf{F}^{\mathrm{p}}_n, \tag{4.90}$$

$$\alpha = \alpha_n + \sqrt{\frac{2}{3}} \Delta\lambda^{\mathrm{p}}, \tag{4.91}$$

$$d = d_n + \Delta\lambda^{\mathrm{d}} \hat{f}^{\alpha}(\alpha). \tag{4.92}$$

Therein, $\boldsymbol{\nu}_0 = \mathrm{dev}(\boldsymbol{\Sigma})/\|\mathrm{dev}(\boldsymbol{\Sigma})\|$ represents the plastic flow direction of the intermediate Mandel stresses $\boldsymbol{\Sigma}$ and $\Delta\lambda^{\bullet} = \Delta t \lambda^{\bullet}$ are the incremental Lagrange multipliers.

Remark 4.2 For readability reasons, the subscript $n+1$ is omitted, e.g. $\alpha = \alpha_{n+1}$. Variables from previous time step are still denoted with a subscript n, e.g. α_n, to allow specific differentiation if they are required for numerical purposes.

Since the iterative data from a previous time step is required to update the internal variables (4.90)-(4.92), they have to be stored to be accessed throughout consecutive iterations. Hence, they are captured in a field of history variables $\mathbf{h}$ in vector form, which encompasses

$$\mathbf{h} = \left\{ \mathbf{C}^{\mathrm{p}-1}_n, \alpha, d \right\}. \tag{4.93}$$

Notably, the inverse of the plastic right Cauchy-Green deformation tensor $\mathbf{C}^{\mathrm{p}-1}$ is stored instead of the plastic deformation gradient $\mathbf{F}^{\mathrm{p}}$. Since the intermediate plastic configuration is only defined up to a rigid body motion, cf. [145, 205], utilisation of the plastic deformation gradient is not feasible. Furthermore, using $\mathbf{C}^{\mathrm{p}-1}$ has benefits in relation to the algorithmic treatment by exploiting relation (4.20). In (4.90), the plastic flow direction of the intermediate Mandel stresses are required, however all other quantities are computed in the principal spatial space. Computation of $\mathrm{dev}(\mathbf{m}) = \mathbf{m}^{\mathrm{iso}}$ in a spectral decomposition and the plastic flow direction $\boldsymbol{\nu}$ for the spatial Mandel stresses, yields

$$\boldsymbol{\nu} = \sum_{i=1}^{3} \nu_i \, \mathbf{n}_i \otimes \mathbf{n}_i, \quad \nu_i = \frac{\varepsilon_i^{\mathrm{e,iso}}}{\sqrt{\sum_{n=1}^{3} \left[\varepsilon_n^{\mathrm{e,iso}}\right]^2}}. \tag{4.94}$$

The principal directions $\mathbf{n}_i$ therein coincide with the principal directions of $\mathbf{b}^{\mathrm{e}}$ since isotropic plasticity is modelled, and therefore also coincide with the principal directions of $\mathbf{F}^{\mathrm{e}}$. The transformation of the flow directions is computed via $\boldsymbol{\nu} = \mathbf{F}^{\mathrm{e}} \cdot \hat{\boldsymbol{\nu}} \cdot \mathbf{F}^{\mathrm{e}-1}$. Using this, and the concept within the return-mapping algorithm and its trial step, allows update of the left Cauchy-Green tensor, such that

$$\mathbf{b}^{\mathrm{e}} = \mathbf{F}^{\mathrm{e,tr}} \cdot \exp\left(-2\Delta\lambda^{\mathrm{p}} \boldsymbol{\nu} \hat{f}^{\boldsymbol{m}}(d)\right) \cdot \left[\mathbf{F}^{\mathrm{e,tr}}\right]^{t}. \tag{4.95}$$

This equation is used for the following implementation and with (4.20) to update the internal variable $\mathbf{C}^{\mathrm{p}-1}$.

Within the local iteration, a two-step algorithm known as the return-mapping algorithm, is implemented. In this algorithm, each local initial computed step at t_{n+1}, denoted as the trial state, is assumed to be elastic, utilising the stored set of history variables $\mathbf{h}_n$ from time step t_n. This is depicted in Fig. 4.2. Since for the first load step, i.e. t_1, some set of initial values for t_0 are required, the set of internal variables $\mathbf{h}_0$ has to be initialised. Assuming no initial plasticity and no pre-damage effects have occurred in the material, these can be initialised to

$$\mathbf{C}_0^{\mathrm{p}-1} = \mathbf{I}, \quad \alpha_0 = 0, \quad d_0 = 0. \tag{4.96}$$

Since the computations from here on out are done in the principal space, only the the eigenvalues are necessary. These are stored in n-tuples, which are indicated with $\bar{\bullet}$. The tensor quantities are recovered using their eigenvectors within the spectral decomposition in Fig. 4.2 step 7 and onward. The quantities of the elastic predictor step read

$$\alpha^{\mathrm{tr}} = \alpha_n, \tag{4.97}$$

$$d^{\mathrm{tr}} = d_n, \tag{4.98}$$

$$\mathbf{C}^{\mathrm{p}-1,\mathrm{tr}} = \mathbf{C}_n^{\mathrm{p}-1} \tag{4.99}$$

$$\mathbf{b}^{\mathrm{e,tr}} = \mathbf{F} \cdot \mathbf{C}_n^{\mathrm{p}-1} \cdot \mathbf{F}^{t} = \sum_{i=1}^{3} \left[\lambda^{\mathrm{e,tr}}\right]^2 \mathbf{n}_i \otimes \mathbf{n}_I, \tag{4.100}$$

$${\varepsilon_i}^{\mathrm{e,tr}} = \ln(\lambda_i^{\mathrm{e,tr}}), \tag{4.101}$$

$$\bar{\mathbf{m}}^{\mathrm{tr,iso}} = 2G f^{\mathrm{iso}}(d_n) \left[\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}} - \frac{\varepsilon^{\mathrm{e,vol}}}{3} \bar{\mathbf{I}}\right], \tag{4.102}$$

where $\bar{\mathbf{I}}$ is the corresponding identity tensor for the principal space and $\boldsymbol{\varepsilon}^{\mathrm{e,tr}}$ is computed using the trial strains and (4.53). Since the plastic deformation is assumed to be isochoric in nature, the volumetric part of the strain decomposition remains constant throughout the local iteration, i.e. $\varepsilon^{\mathrm{e,vol}} = \varepsilon^{\mathrm{e,tr,vol}}$. Using these variables, trial values for both dissipation potentials (4.61) and (4.62) are computed, i.e.

$$\Phi^{\mathrm{p,tr}} = \|(\bar{\mathbf{m}}^{\mathrm{tr,iso}})\| \hat{f}^{\boldsymbol{m}}(d_n) - \sqrt{\frac{2}{3}} \left[\sigma_{\mathrm{y}0} - h\, \alpha_n^{n_{\mathrm{p}}}\right], \tag{4.103}$$

$$q^{\mathrm{tr}} = -\frac{1}{2} K \left[f^{\mathrm{vol}}(d_n)\right]' \left[\varepsilon^{\mathrm{e,vol}}\right]^2 - G \left[f^{\mathrm{iso}}(d_n)\right]' \|\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso,tr}}\|^2 - \beta_{\mathrm{d}} \left[\phi - d_n\right] \tag{4.104}$$

$$\Phi^{\mathrm{d,tr}} = q^{\mathrm{tr}} \hat{f}^{\alpha}(d_n) - q_{\min} \left[1 - f^{q}(d_n)\right]^{n_{\mathrm{d}}}. \tag{4.105}$$

If neither of those trial quantities are greater than zero, and with the initialised Lagrange parameters $\Delta\lambda^{\mathrm{p}} = \Delta\lambda^{\mathrm{d}} = 0$, the KKT (4.67) and (4.68) are inherently fulfilled and the assumed elastic case is indeed a valid solution. In this case, the trial quantities are solutions of the local problem and step 4 in Fig. 4.2 can be skipped. The stresses and driving forces can be simply updated in a straightforward fashion, see Fig. 4.2. These are consequently used for the computation of the tangent computations, see Fig. 4.4. No further local iteration is required. However, if one of those trial values for the dissipations are greater than zero, a local update of the Lagrangian multipliers is necessary, as the load step is not elastic.

Remark 4.3 From this point onward, also regarding details in Appendix A and Appendix B, only the coupled case of damage and plasticity is considered, i.e. $\Phi^{\mathrm{p,tr}} > 0$ and $\Phi^{\mathrm{d,tr}} > 0$. In theory, the uncoupled case can occur, where only plasticity or damage might evolve. However, this specifically not considered in the derivation here. The case distinctions and presentation of all additional derivatives would drastically increase the length of this work and might lead to confusion. However, understanding the concepts of the presented work allows the derivation of the not specifically mentioned cases.

A non-elastic step means evolution of damage and plasticity and therefore the internal variables have to be updated accordingly. To fulfil the KKT conditions, the Lagrangian multipliers have to be adjusted. In this implementation, it is achieved with an active-set algorithm, cf. [151, 162, 206], and its implementation depicted in Fig. 4.3. For multi-surface problems, other algorithms can be used as well, like the Fischer-Burmeister and other NCP methods, cf. [128].

The active set is initialised with $k = 0$ as $\mathcal{A}_0 = \emptyset$. In each iteration the active set is filled with the parts of the total set $\mathcal{P} = \{\mathrm{p,d}\}$, which violates the constraints the most, i.e. $\mathcal{A}_{k+1} = \{p\}$ if $\Phi^{\mathrm{p}} > \Phi^{\mathrm{d}} \geq 0$, or $\mathcal{A}_{k+1} = \{d\}$ if $\Phi^{\mathrm{d}} > \Phi^{\mathrm{p}} \geq 0$. In the initial case $k = 1$, these quantities correlate with the trial values, i.e. $\Phi^{\mathrm{p}} = \Phi^{\mathrm{p,tr}}$ and $\Phi^{\mathrm{d}} = \Phi^{\mathrm{d,tr}}$. For the active set, the corresponding Lagrangian multipliers are solved using a local Newton-Raphson iteration, with

$$\boldsymbol{\Lambda} = \begin{bmatrix} \Delta\Lambda^{\mathrm{p}} \\ \Delta\Lambda^{\mathrm{d}} \end{bmatrix}, \quad \mathbf{r}^{\mathrm{loc}} = \begin{bmatrix} \Phi^{\mathrm{p}} \\ \Phi^{\mathrm{d}} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \dfrac{\partial \Phi^{\mathrm{p}}}{\partial \Delta\lambda^{\mathrm{p}}} & \dfrac{\partial \Phi^{\mathrm{p}}}{\partial \Delta\lambda^{\mathrm{d}}} \\ \dfrac{\partial \Phi^{\mathrm{d}}}{\partial \Delta\lambda^{\mathrm{p}}} & \dfrac{\partial \Phi^{\mathrm{d}}}{\partial \Delta\lambda^{\mathrm{d}}} \end{bmatrix}. \tag{4.106}$$

Therein, $\boldsymbol{\Lambda}$ is the unknown update for the Lagrangian multipliers, $\mathbf{r}^{\mathrm{loc}}$ the local residual and $\mathbf{J}$ the corresponding Jacobian, which is detailed in Appendix A.2. Each iteration yields an update for the Lagrangian multipliers

$$\Delta\lambda^{\mathrm{p}} = \Delta\lambda^{\mathrm{p}}_k + \Delta\Lambda^{\mathrm{p}}, \tag{4.107}$$

$$\Delta\lambda^{\mathrm{d}} = \Delta\lambda^{\mathrm{d}}_k + \Delta\Lambda^{\mathrm{d}}, \tag{4.108}$$

with

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_k + \Delta\boldsymbol{\Lambda}, \tag{4.109}$$

$$\Delta\boldsymbol{\Lambda} = -\mathbf{J}^{-1} \cdot \mathbf{r}^{\mathrm{loc}}. \tag{4.110}$$

This local update is computed until the change in update is sufficiently small, i.e. $\|\Delta\boldsymbol{\Lambda}\| \leq \mathrm{tol}$ with a predefined tolerance, which is set to $\mathrm{tol} = 1 \times 10^{-12}$ in this implementation.

To update the active set, in each iteration the active set from the previous iteration k is used as the initial set for $k+1$, i.e. $\mathcal{A}_{k+1} = \mathcal{A}_k$. This set is updated by removing the smallest incremental Lagrange multiplier j and, since it might be negative due to numerical tolerances, is reset to zero, i.e.

$$\mathcal{A}_{k+1} = \mathcal{A}_{k+1} \setminus \{j\}, \quad \Delta\lambda^j = 0, \quad \text{with} \quad \Delta\lambda^j = \min\left\{\Delta\lambda^j \in \mathcal{A}_k \,|\, \Delta\lambda^j < 0\right\}. \tag{4.111}$$

Afterwards, the largest, yet unconsidered constraint j which exceeds their bound is added to the active-set, i.e.

$$\mathcal{A}_{k+1} = \mathcal{A}_{k+1} \cup \{j\}, \quad \text{with} \quad \Phi^j = \max\left\{\Phi^j \in \mathcal{P} \setminus \mathcal{A}_k \,|\, \Phi^j > 0\right\}. \tag{4.112}$$

After these adjustment, if the active set does not change, i.e. $\mathcal{A}_{k+1} = \mathcal{A}_k$ still holds, the algorithm is finished and the update for the Lagrangian multipliers computed, which fulfil the KKTs. Otherwise, the algorithm continues until $k = k_{\max}$, which limits the maximum number of local iterations. If this occurs, the local iteration is stopped and the current global iteration is cancelled as well, since the local material model does not converge for the current load increment. A time step adjustment is implemented, which alters the global load and tries the boundary value problem (BVP) again for a reduce load step.

This time step size control is required, as the complexity of the material model might reach load increments, where a too large step size might not reach a converging solution in some local element iteration. In this case, the global load application is reduced and the global BVP computed with smaller load increments. This is more beneficial than initially setting a constant, very small load increment, as this would require this load step to be applied throughout the complete simulation. Furthermore, the automatic step size control is implemented, such that time steps can be increased, if the global problem converges quickly again, overall speeding up the computation time.

Local damage model

Input: $\mathbf{F}$,ϕ,$\nabla_{\mathbf{X}}\phi$, and previous history field $\mathbf{h}_n=\left\{\mathbf{C}_n^{\mathrm{p}-1},\alpha_n,d_n\right\}$

1. Compute trial left Cauchy-Green tensor and spectral decomposition for trial Hencky strains

$$\mathbf{b}^{\mathrm{e,tr}} = \mathbf{F}\cdot\mathbf{C}_n^{\mathrm{p}-1}\cdot\mathbf{F}^t = \sum_{i=1}^{3}\left[\lambda^{\mathrm{e,tr}}\right]^2\mathbf{n}_i\otimes\mathbf{n}_i, \quad {\varepsilon_i}^{\mathrm{e,tr}} = \ln(\lambda_i^{\mathrm{e,tr}})$$

2. Compute spatial return direction $\boldsymbol{\nu} = \|\boldsymbol{\varepsilon}^{\mathrm{e,iso,tr}}\|^{-1}\boldsymbol{\varepsilon}^{\mathrm{e,iso,tr}}$
3. Evaluate trial values of dissipation potentials

$$\begin{aligned} \Phi^{\mathrm{p,tr}} &= \|\bar{\mathbf{m}}^{\mathrm{tr,iso}}\|\hat{f}^{\boldsymbol{m}}(d_n) - \sqrt{\frac{2}{3}}\left[\sigma_{\mathrm{y}0} - h\,\alpha_n^{n_\mathrm{p}}\right] \\ q^{\mathrm{tr}} &= -\frac{1}{2}K\left[f^{\mathrm{vol}}(d_n)\right]'\left[\varepsilon^{\mathrm{e,vol}}\right]^2 - G\left[f^{\mathrm{iso}}(d_n)\right]'\|\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso,tr}}\|^2 - \beta_\mathrm{d}\left[\phi - d_n\right] \\ \Phi^{\mathrm{d}} &= q^{\mathrm{tr}}\hat{f}^{\alpha}(d_n) - q_{\min}\left[1 - f^q(d_n)\right]^{n_\mathrm{d}}\,. \end{aligned}$$

4. Initialise Lagrangian multipliers $\Delta\lambda^{\mathrm{p}} = \Delta\lambda^{\mathrm{d}} = 0$
5. If $\Phi^{\mathrm{p,tr}} > 0$ or $\Phi^{\mathrm{d,tr}} > 0$: Update Lagrange multipliers ⟶ Fig. 4.3
6. Update internal variables and elastic left Cauchy-Green tensor

$$\begin{aligned} \alpha &= \alpha_n + \sqrt{\frac{2}{3}}\Delta\lambda^{\mathrm{p}} \\ d &= d_n + \Delta\lambda^{\mathrm{d}}\hat{f}^{\alpha}(\alpha) \\ \mathbf{b}^{\mathrm{e}} &= \sum_{i=1}^{3}\exp\left(2\,{\varepsilon_i}^{\mathrm{e,tr}} - 2\,\Delta\lambda^{\mathrm{p}}\hat{f}^{\boldsymbol{m}}(d)\nu_i\right)\mathbf{n}_i\otimes\mathbf{n}_i \\ \mathbf{C}^{\mathrm{p}-1} &= \mathbf{F}^{-1}\cdot\mathbf{b}^{\mathrm{e}}\cdot\mathbf{F}^{-t} \end{aligned}$$

7. Compute derivatives for tangent contributions (see Appendix A.1)
8. Compute stresses and driving forces

$$\begin{aligned} \mathbf{m} &= \sum_{i=1}^{3} m_i\,\mathbf{n}_i\otimes\mathbf{n}_i, \quad \text{with} \quad m_i = K f^{\mathrm{vol}}(d)\,\varepsilon^{\mathrm{e,vol}} + 2\,G f^{\mathrm{iso}}(d)\,{\varepsilon_i}^{\mathrm{e,iso}} \\ \mathbf{P}^{\mathrm{K}} &= \mathbf{m}\cdot\mathbf{F}^{-t} \\ Y &= \beta_\mathrm{d}\left[\phi - d\right] \\ \mathbf{Y} &= c_\mathrm{d}\nabla_{\mathbf{X}}\phi \end{aligned}$$

9. Compute tangent contributions ⟶ Fig. 4.4

Figure 4.2: Return-mapping algorithm for the local damage model.

```
Active-set procedure

Initialise: $k{=}0$, $\mathcal{A}_0{=}\{\emptyset\}$, $\varPhi^{\mathrm{p}}{=}\varPhi^{\mathrm{p,tr}}$, $\varPhi^{\mathrm{d}}{=}\varPhi^{\mathrm{d,tr}}$

1. Update active set $\mathcal{A}_{k+1} = \mathcal{A}_k$
    a) If $k = k_{\max} \longrightarrow$ Exit active-set environment and reduce global load step
    b) Remove smallest active multiplier for inactive constraint $\varPhi^{\bullet} \leq 0$ from $\mathcal{A}_{k+1}$
    c) Add largest inactive constraint with active $\varPhi^{\bullet} \geq 0$ to $\mathcal{A}_{k+1}$
    d) If $\mathcal{A}_{k+1} = \mathcal{A}_k \longrightarrow$ Exit active-set environment
    e) Set $k = k + 1$
2. Update active Lagrangian multiplier with local Newton-Raphson scheme
    a) Compute local Jacobian $\mathbf{J}$ (see Appendix A.2)
    b) Update active Lagrangian multipliers
           $\Delta\boldsymbol{\Lambda} = -\mathbf{J}^{-1} \cdot \mathbf{r}^{\mathrm{loc}}$
           $\Delta\lambda^{\mathrm{d}} = \Delta\lambda^{\mathrm{d}}_k + \Delta\Lambda^{\mathrm{p}}$
           $\Delta\lambda^{\mathrm{d}} = \Delta\lambda^{\mathrm{d}}_k + \Delta\Lambda^{\mathrm{d}}$
    c) Update variables
           $\alpha = \alpha_n + \sqrt{\frac{2}{3}}\Delta\lambda^{\mathrm{p}}$
           $d = d_n + \Delta\lambda^{\mathrm{d}}\hat{f}^{\alpha}(\alpha)$
           $\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso}} = \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso,tr}} - \Delta\lambda^{\mathrm{p}}\,{}^{\bar{\circ}\;\mathrm{tr}}\hat{f}^{\boldsymbol{m}}(d)$
           $q = -\frac{K}{2}[f^{\mathrm{vol}}(d)]'[\varepsilon^{\mathrm{e,vol}}]^2 - G\,[f^{\mathrm{iso}}(d)]'\|\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso}}\|^2 + \beta_{\mathrm{d}}[\phi - d]$
    d) Compute local residual values
           $r^{\mathrm{loc}}_1 = \varPhi^{\mathrm{p}} = 2\,Gf^{\mathrm{iso}}(d)\hat{f}^{\boldsymbol{m}}(d)\|\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso}}\| - \sqrt{\frac{2}{3}}[\sigma_{\mathrm{y0}} + h\,\alpha^{n_{\mathrm{p}}}]$   if p $\in \mathcal{A}_k$
           $r^{\mathrm{loc}}_2 = \varPhi^{\mathrm{d}} = q\,\hat{f}^{\alpha}(\alpha) - q_{\min}\,[1 - f^{q}(d)]^{n_{\mathrm{d}}}$   if d $\in \mathcal{A}_k$
    e) If $\|\Delta\Lambda\| \geq$ tol $\longrightarrow$ goto 2. a)
3. Additional active-set iteration required $\longrightarrow$ goto 1.
```

Figure 4.3: Active-set algorithm for the computation of the Lagrangian multipliers.

Tangent contributions

1. Compute tangent operator $\boldsymbol{c}$

$$a_{ij} = \begin{cases} -m_i & i = j \\ \dfrac{m_i[\lambda_j^{\mathrm{e}}]^2 - m_k[\lambda_i^{\mathrm{e}}]^2}{[\lambda_i^{\mathrm{e}}]^2 - [\lambda_j^{\mathrm{e}}]^2} & i \neq j \text{ and } \lambda_i^{\mathrm{e}} \neq \lambda_j^{\mathrm{e}} \\ \dfrac{[\lambda_j^{\mathrm{e}}]^2}{[\lambda_i^{\mathrm{e}}]^2}\left[\frac{1}{2}\frac{\partial m_i}{\partial \varepsilon_i} - m_i\right] - \frac{1}{2}\frac{\partial m_j}{\partial \varepsilon_i} & i \neq j \text{ and } \lambda_i^{\mathrm{e}} = \lambda_j^{\mathrm{e}} \end{cases}$$

$$\mathbb{c} = \sum_{i=1}^{3}\sum_{j=1}^{3}\Bigg[\frac{\partial m_i}{\partial \varepsilon_j}\mathbf{n}_i \otimes \mathbf{n}_i \otimes \mathbf{n}_j \otimes \mathbf{n}_j \\ + a_{ij}\left[\mathbf{n}_i \otimes \mathbf{n}_j \otimes \mathbf{n}_i \otimes \mathbf{n}_j + \mathbf{n}_i \otimes \mathbf{n}_j \otimes \mathbf{n}_j \otimes \mathbf{n}_i\right]\Bigg]$$

2. Compute derivatives of $\mathbf{P}^{\mathrm{K}}$ w.r.t. $\mathbf{F}$ and ϕ

$$\frac{\partial \mathbf{P}^{\mathrm{K}}}{\partial \mathbf{F}} = \left[\mathbf{I}\,\overline{\otimes}\,\mathbf{F}^{-1}\right] : \mathbb{c} \cdot \mathbf{F}^{-t} + \mathbf{I}\,\overline{\otimes}\left[\mathbf{F}^{-1} \cdot \mathbf{P}^{\mathrm{K}}\right]$$

$$\frac{\partial \mathbf{P}^{\mathrm{K}}}{\partial \phi} = \frac{\partial \mathbf{m}}{\partial \phi} \cdot \mathbf{F}^{-t} = \left[\sum_{i=1}^{3}\frac{\partial m_i}{\partial \phi}\,\mathbf{n}_i \otimes \mathbf{n}_i\right] \cdot \mathbf{F}^{-t}$$

3. Compute derivatives of Y w.r.t $\mathbf{F}$ and ϕ

$$\frac{\partial Y}{\partial \mathbf{F}} = \left[\sum_{i=1}^{3}\frac{\partial Y}{\partial \varepsilon_i}\frac{1}{2[\lambda_i^{\mathrm{e}}]^2}\mathbf{n}_i \otimes \mathbf{n}_i\right] : \left[\mathbf{I}\,\overline{\otimes}\,\mathbf{F} \cdot \mathbf{C}_n^{\mathrm{p}-1} + \mathbf{F} \cdot \mathbf{C}_n^{\mathrm{p}-1}\,\underline{\otimes}\,\mathbf{I}\right]$$

$$\frac{\partial Y}{\partial \phi} = \beta_d\left[1 - \frac{\partial d}{\partial \phi}\right]$$

4. Compute derivative of $\mathbf{Y}$ w.r.t. $\nabla_{\boldsymbol{X}}\phi$

$$\frac{\partial \mathbf{Y}}{\partial \nabla_{\boldsymbol{X}}\phi} = c_{\mathrm{d}}\,\mathbf{I}$$

Figure 4.4: Specific computation of all tangent contributions. Non-specified derivatives are found in Appendix A.

Chapter 5

Sensitivity analysis for ductile damage materials

This chapter describes the challenging task of enhancing the material model presented in Chap. 4 with sensitivity information to enable gradient-based optimisation in Chap. 6. A variational approach is utilised to derive gradients for geometry as the chosen design variables. For the derivation of the geometrical gradients, an enhanced kinematic concept is depicted. The derivation of the gradients is initially applied to the continuous equations and discretised later. Due to the dissipative material model considered, history fields have to be included in the gradient derivation and the numerics. Additionally, the above described developments are further used to derive analytical sensitivities w.r.t. the constitutive material parameters as the design variables. For the purpose of easy readability, a large number of detailed derivations and resulting derivatives are not specified in this chapter and can be found in Appendix B. Remarks and an overview for the implementation into AOPS given and results from a simple benchmark example presented.

5.1 Design sensitivity analysis

As presented in Chap. 3, gradient-based optimisation techniques are generally the most efficient methods when dealing with mathematical optimisation problems. Considering the underlying complexity of the ductile damage material and the numerical effort in simulating its physical responses, gradient-based optimisation is best used to efficiently solve the arising optimisation problems in Chap. 6. However, to employ these solution techniques, gradient information have to be provided. In Sect. 3.4, possible approaches to derive these gradient information have been briefly presented. The easiest approach, by utilising the finite difference method (FDM), comes with a high cost in computational effort. Since for each design variable at least one additional computation of the boundary value problem (BVP) is required (or two if the central difference quotient (CDQ) is computed), this method is not feasible if many design variables are considered with the material model at hand. The applied method for this work is the analytical derivation of gradient by means of the variational sensitivity analysis approach, i.e. the analytical derivation of the gradients on the continuous level and later implementation into the numerical framework.

The ductile damage material model in Chap. 4, from the viewpoint of continuum mechanics, has been derived on a continuous basis. The discretisation for its application within the numerical setting has been performed afterwards. As described in Sect. 3.4.1, two approaches to derive the gradient information for optimisation can be followed. Here, the choice of utilising the continuous functionals of the material model, i.e. the quantities prior to discretisation, as the starting point of analytical sensitivity analysis are chosen. This requires a deeper investigation to derive the quantities, but is more computationally efficient. This is very important, due to the already high computational cost of the coupled material model. The continuous equations are used as the starting point and a total variation of these allows computation of the continuous gradients. While this method is mathematically very complex, the resulting gradients are available prior to applying a discretisation scheme. In return, the choice of the applied discretisation technique can be made freely afterwards, e.g. using a finite element or isogeometric discretisation method. Finally, after a consistent discretisation, the resulting discrete equations have to be implemented. Since the same discretisation technique is chosen for both the structural analysis and the sensitivity analysis, respectively, this can be taken into account for an efficient implementation into the computational environment.

The variational approach utilised in this thesis is based on the conceptual ideas presented in [17, 15]. This approach enhances the continuum mechanical descriptions and therefore can be directly applied to the mechanical problems at hand. In standard nonlinear kinematics, the deformation is only dependent on the referential geometry $\boldsymbol{X}$ and time t. Since therein the initial geometry is assumed fixed, changes can only be induced due to loads which are applied throughout time t. From the viewpoint of structural optimisation (SO) however, the design no longer is fixed. As an essential component of SO, it is allowed to change throughout the optimisation. Therefore, the initial geometry is dependent on the design variables. The enhanced conceptual approach in [17, 15] introduces a new manifold, where the reservoir of material resides, which can be mapped to a new referential configuration or the resulting spatial configuration. This is denoted as the *enhanced kinematics* in the following.

The original work considered elastic problems. However, this has to be extended to include the dissipative behaviour of ductile damage within this work. The main challenges arsing in this case are a) the additional nonlocal field variable in form of nonlocal damage ϕ and its global balance equations, and b) the introduced history variables to capture the previous dissipative load steps.

Problem a) is comparably straightforward to consider, due to the thorough foundation provided in the previous chapter. From this, the two weak forms of the described material model have to be enhanced by means of a sensitivity analysis. Hence, one equation describes the mechanical problem, i.e. deformation $\boldsymbol{\varphi}$, and the other the damage problem, i.e. nonlocal damage ϕ. Consequently, all contributions within this work will consist of a mechanical and a damage part. To the best knowledge of the author, this kind of research, i.e. analytical sensitivity analysis of regularised nonlocal damage, is novel and has yet to be conducted in other mediums of literature.

Problem b), i.e. the consideration of history variables, has been successfully dealt with in literature, mainly in the form of consideration of plastic effects, cf. [145, 146, 147, 233, 234]. The same concept is applied here to derive the history dependent variations and its implementation into the numerical environment. However, in those works only elasticity and elasto-plasticity had to be considered. In this work now the elastic, elasto-plastic, elasto-damage and elasto-plastic damage, i.e. ductile damage, case can occur during the loading. Therefore, they all have to be considered in the derivation and subsequent implementation. In addition, the underlying material model follows a different structure, which requires additional special care regarding the consideration of history variables. Generally, the material model is more complex in nature, making derivation of gradient information more taxing and not straightforward. Hence, most of the derivations of these detailed derivatives is not specified in their respective sections and instead moved to Appendix B to ease with readability of this chapter.

Some important aspects regarding these two problems and sensitivity analysis for nonlocal damage have already been published by the author in [96, 98, 99, 100, 103].

Since the applied approach directly derives the gradient information for the continuous description, the resulting quantities have to be discretised and implemented into the finite element (FE) environment afterwards, see e.g. [37, 120, 238] for literature on finite element methods. Since the underlying geometric description will be generated by means of Computer Aided Geometric Design (CAGD), cf. [74], this has to be further taken into account for generation of new geometric shapes. Of special note here is the resulting adaptations to a standard optimisation framework required for the computation of history dependent sensitivities, cf. [99, 145, 147, 233].

5.2 Decoupling of implicit dependencies

In standard, nonlinear kinematics, the reference body $\mathcal{B}_0$ is assumed to be fixed. Application of SO however, introduces a variable reference configuration within each optimisation step. For the derivation of the sensitivities it is therefore beneficial to enhance the viewpoint

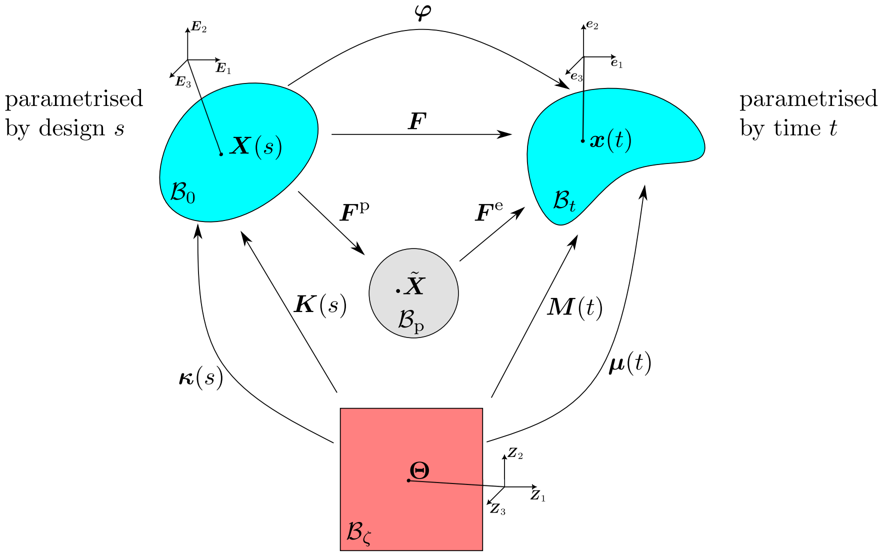


Figure 5.1: Enhanced kinematic concept.

of the underlying kinematics by the so-called *intrinsic formulation*, cf. [17, 15]. Beyond the original works, this method has already been successfully employed in many different fields. Examples include elasto-plasticity [145], multi-scale physics and FE^2 methods [129], shell structures and singular value decomposition [86], isotropic damage [18], or in porous media [232], to name a few.

This concept introduces an additional domain $\mathcal{B}_\zeta$ with the intrinsic coordinate $\boldsymbol{\Theta} \in \mathcal{B}_\zeta$, see Fig. 5.1. Exploitation of the (sufficiently smooth) mapping between the three configurations allows the calculation of derivatives w.r.t. the current configuration, without any dependencies of the referential configuration. The additional mappings read

$$\boldsymbol{\kappa} : \begin{cases} \mathcal{B}_\zeta \longrightarrow \mathcal{B}_0 \\ (\boldsymbol{\Theta},s) \mapsto \boldsymbol{X} = \boldsymbol{\kappa}(\boldsymbol{\Theta},s) \end{cases}, \tag{5.1}$$

and

$$\boldsymbol{\mu} : \begin{cases} \mathcal{B}_\zeta \longrightarrow \mathcal{B}_t \\ (\boldsymbol{\Theta},t) \mapsto \boldsymbol{x} = \boldsymbol{\mu}(\boldsymbol{\Theta},t) \end{cases}, \tag{5.2}$$

for any time t and design s. Here, the design s is introduced as a general, scalar design variables. It parametrises the reference body such that $\mathcal{B}_0 = \mathcal{B}_0(s)$. The above mappings allow the deformation map $\boldsymbol{\varphi}$ in (4.1) to be rewritten as a composition, i.e.

$$\boldsymbol{\varphi} = \boldsymbol{\mu} \circ \boldsymbol{\kappa}^{-1} \quad \text{with} \quad \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}) = (\boldsymbol{\mu} \circ \boldsymbol{\kappa}^{-1})(\boldsymbol{X}) = \boldsymbol{\mu}(\boldsymbol{\kappa}^{-1}(\boldsymbol{X})), \tag{5.3}$$

where $\circ$ is the classically used symbol in mathematics denoting the composition of the two mappings.

The additional operations

$$\mathrm{GRAD}(\bullet) = \nabla_{\boldsymbol{\Theta}}(\bullet) \quad \text{and} \quad \mathrm{DIV}(\bullet) = \nabla_{\boldsymbol{\Theta}} \cdot (\bullet), \tag{5.4}$$

can be defined, which allow derivation of the counterparts to the deformation gradient and the Jacobian, i.e.

$$\begin{aligned} \boldsymbol{K} &:= \mathrm{GRAD}(\boldsymbol{\kappa}) = \nabla_{\boldsymbol{\Theta}}\boldsymbol{\kappa} = \nabla_{\boldsymbol{\Theta}}\boldsymbol{X} \quad \text{and} \quad J_{\mathrm{K}} = \det(\boldsymbol{K}), \\ \boldsymbol{M} &:= \mathrm{GRAD}(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\Theta}}\boldsymbol{\mu} = \nabla_{\boldsymbol{\Theta}}\boldsymbol{x} \quad \text{and} \quad J_{\mathrm{M}} = \det(\boldsymbol{M}). \end{aligned} \tag{5.5}$$

Similarly to the definitions of deformation gradient $\boldsymbol{F}$ in Sect. 4.2, no specific distinction between the functions and their function values is made in the derivations of (5.5) above.

Consequently, volume integrals can be mapped between the different configurations, such that

$$\int_{\mathcal{B}_0} \mathrm{d}V = \int_{\mathcal{B}_t} \frac{1}{J}\,\mathrm{d}v = \int_{\mathcal{B}_\zeta} J_{\mathrm{K}}\,\mathrm{d}V_\zeta. \tag{5.6}$$

Using (5.5), it is possible to multiplicatively decompose the deformation gradient, i.e.

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\Theta}} \cdot \frac{\partial \boldsymbol{\Theta}}{\partial \boldsymbol{X}} = \boldsymbol{M} \cdot \boldsymbol{K}^{-1}. \tag{5.7}$$

With the above concept, the following variations can be defined

$$\delta \boldsymbol{X}(s) = \delta \boldsymbol{\kappa}(\boldsymbol{\Theta},s) \quad \text{and} \quad \delta \boldsymbol{x}(t) = \delta \boldsymbol{\mu}(\boldsymbol{\Theta},t) \tag{5.8}$$

and consequently

$$\delta \boldsymbol{K} = \mathrm{GRAD}(\delta \boldsymbol{\kappa}) \quad \text{and} \quad \delta \boldsymbol{M} = \mathrm{GRAD}(\delta \boldsymbol{\mu}), \tag{5.9}$$

which allows the derivation of the total variation of the deformation gradient, exploiting (5.7), such that

$$\begin{aligned} \delta \boldsymbol{F} = \delta \mathrm{Grad}(\boldsymbol{x}) &= \mathrm{Grad}(\delta \boldsymbol{x}) - \mathrm{Grad}(\boldsymbol{x}) \cdot \mathrm{Grad}(\delta \boldsymbol{X}) \\ &= \mathrm{Grad}(\delta \boldsymbol{u}) - \mathrm{Grad}(\boldsymbol{u}) \cdot \mathrm{Grad}(\delta \boldsymbol{X}). \end{aligned} \tag{5.10}$$

For more details, see [17, 15, 130].

5.3 Continuous ductile damage sensitivities

To derive the sensitivities for the previously described ductile damage material model in Chap. 4, a variational approach is applied. While in the environment of the structural analysis, only the field variables $\boldsymbol{w}$ and the history field $\boldsymbol{h}$ are allowed to change, for the optimisation this now extends to an additional change in design $\boldsymbol{s}$. Here, the introduction of a bold faced design $\boldsymbol{s}$, in contrast to the scalar s previously, is motivated by its vector form later on in the implementation of these variables in a discrete setting. The same motivation is used for the quantity $\boldsymbol{w} = \{\boldsymbol{\varphi},\phi\}$ and the history field $\boldsymbol{h} = \left\{\boldsymbol{C}^{\mathrm{p}-1},\alpha,d\right\}$. According to the introduced notation in Sect. 2.1, both these quantities should described a vector with a basis system. However, both $\boldsymbol{w}$ and $\boldsymbol{h}$ are fields which capture certain sets of variables which are tensor and scalar valued. This is an exception to the previously described notation and motivated by the similar utilisation in the discrete environment. In that setting, $\mathbf{w}$ is the vector of field variables, see Sect. 4.6.1, and the discrete history field $\mathbf{h}$ is introduced in (4.93). This helps with a consistent denotation of the quantities within this chapter. Hence, their variations can be stated as

$$\delta \boldsymbol{w} = \{\delta \boldsymbol{\varphi}, \delta \phi\} \quad \text{and} \quad \delta \boldsymbol{h} = \left\{\delta \boldsymbol{C}^{\mathrm{p}-1}, \delta \alpha, \delta d\right\}. \tag{5.11}$$

Following the same idea, the test functions from Sect. 4.4.2 are captured identically, i.e. $\boldsymbol{\eta} = \left\{\boldsymbol{\eta}^{\varphi},\eta^{\phi}\right\}$.

Consequently, the total variation of (4.49), with the assumption of design independent external forces ($\boldsymbol{r}^{\mathrm{ext}} = \boldsymbol{0}$), reads

$$\begin{aligned} \delta \boldsymbol{r} &= \delta_{\boldsymbol{w}} \boldsymbol{r} + \delta_{\boldsymbol{s}} \boldsymbol{r} + \delta_{\boldsymbol{h}_n} \boldsymbol{r} \\ &= \boldsymbol{k}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h}_n;\boldsymbol{\eta},\delta \boldsymbol{w}) + \boldsymbol{p}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h}_n;\boldsymbol{\eta},\delta \boldsymbol{s}) + \boldsymbol{h}^{\boldsymbol{w}}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{h}_n;\boldsymbol{\eta},\delta \boldsymbol{h}_n) = \boldsymbol{0}. \end{aligned} \tag{5.12}$$

Therein, the first partial variation is identified as the (continuous) tangent stiffness $\boldsymbol{k}$. The other two new terms are the (continuous) pseudo load $\boldsymbol{p}$ and the (continuous) history sensitivity $\boldsymbol{h}^{\boldsymbol{w}}$. The variable $\boldsymbol{h}_n$ describes the set of history variables $\boldsymbol{h}$ from the previous time step t_n. This total variation has to vanish to fulfil equilibrium in the newly generated design, as a change in that design must not violate the equilibrium condition (4.49), cf. [17, 103, 146, 233].

5.3.1 Pseudo load

The pseudo load term in (5.12) corresponds with the partial variation of the global weak forms (4.49) w.r.t. an arbitrary design $\boldsymbol{s}$. Since this weak form consists of a global deformation and global damage contribution, the pseudo load terms require this split as well. Using the previously described enhanced kinematic concept, and therefore transferring the integrals into the parameter space $\mathcal{B}_\zeta$, the partial variation for the mechanical part consists of three different partial variations and reads

$$\begin{aligned} p^\varphi = \delta_{\boldsymbol{s}} r^\varphi = \delta_{\boldsymbol{s}} \left[\int_{\mathcal{B}_0} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^\varphi \, \mathrm{d}V \right] & \\ = \int_{\mathcal{B}_\zeta} \delta_{\boldsymbol{s}} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^\varphi \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta & \\ + \int_{\mathcal{B}_\zeta} \boldsymbol{P}^{\mathrm{K}} : \delta_{\boldsymbol{s}} \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^\varphi \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta & \\ + \int_{\mathcal{B}_\zeta} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^\varphi \, \delta_{\boldsymbol{s}} J_{\mathrm{K}} \, \mathrm{d}V_\zeta . & \end{aligned} \tag{5.13}$$

The damage part for an arbitrary design $\boldsymbol{s}$ consists of five partial variations and reads

$$\begin{aligned} p^\phi = \delta_{\boldsymbol{s}} r^\phi = \delta_{\boldsymbol{s}} \left[\int_{\mathcal{B}_0} \boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^\phi \, \mathrm{d}V + \int_{\mathcal{B}_0} Y \eta^\phi \, \mathrm{d}V \right] & \\ = \int_{\mathcal{B}_\zeta} \delta_{\boldsymbol{s}} \boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^\phi \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta & \\ + \int_{\mathcal{B}_\zeta} \boldsymbol{Y} \cdot \delta_{\boldsymbol{s}} \nabla_{\boldsymbol{X}} \eta^\phi \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta & \\ + \int_{\mathcal{B}_\zeta} \boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^\phi \, \delta_{\boldsymbol{s}} J_{\mathrm{K}} \, \mathrm{d}V_\zeta & \\ + \int_{\mathcal{B}_\zeta} \delta_{\boldsymbol{s}} Y \, \eta^\phi \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta & \\ + \int_{\mathcal{B}_\zeta} Y \, \eta^\phi \, \delta_{\boldsymbol{s}} J_{\mathrm{K}} \, \mathrm{d}V_\zeta . & \end{aligned} \tag{5.14}$$

5.3.2 History variation

The underlying material model employs state-dependent history variables to capture previous dissipative load-cases, i.e. the evolution of damage and plasticity. Even though the problems considered in this work are quasi-static in nature, for the evolution equations a time integration has to be employed and the load applied in pseudo-time steps. Consequently, the history variables form the previous time step t_n influence the next time increment t_{n+1}. They are stored after a converged global iteration and provided for the next load step. In turn, this has to be considered in the total variation of the global weak form equilibrium. It is captured in the partial variation w.r.t. the internal history variables of previous time step t_n, i.e.

$$\delta_{\boldsymbol{h}_n} r^{\varphi} = \int_{\mathcal{B}_0} \delta_{\boldsymbol{h}_n} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \, \mathrm{d}V \tag{5.15}$$

for the mechanical part, and

$$\delta_{\boldsymbol{h}_n} r^{\phi} = \int_{\mathcal{B}_0} \delta_{\boldsymbol{h}_n} Y \, \eta^{\phi} \, \mathrm{d}V \tag{5.16}$$

for the damage part. Two things are notable here. First of all, the mapping into the material space is not required for these specific integrals. Since the history variables are only part of the constitutive model, they are not influenced by the geometric description and therefore no implicit dependencies have to be considered which require the enhanced kinematic viewpoint. Secondly, only the scalar valued nonlocal driving force contributes to the history sensitivities. The gradient contribution $\boldsymbol{Y}$ only depends on the material parameter c_{d} and therefore its variation vanishes, see Fig. 4.4, step 4.

Considering the set of internal variables $\boldsymbol{h} = \left\{\boldsymbol{C}^{\mathrm{p}-1}, \alpha, d\right\}$, the variations within the integrals can be further specified as

$$\delta_{\boldsymbol{h}_n} \boldsymbol{P}^{\mathrm{K}} = \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \boldsymbol{h}_n} \cdot \delta \boldsymbol{h}_n = \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} : \delta \boldsymbol{C}_n^{\mathrm{p}-1} + \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \alpha_n} \delta \alpha_n + \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial d_n} \delta d_n, \tag{5.17}$$

$$\delta_{\boldsymbol{h}_n} Y = \frac{\partial Y}{\partial \boldsymbol{h}_n} \cdot \delta \boldsymbol{h}_n = \frac{\partial Y}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} : \delta \boldsymbol{C}_n^{\mathrm{p}-1} + \frac{\partial Y}{\partial \alpha_n} \delta \alpha_n + \frac{\partial Y}{\partial d_n} \delta d_n. \tag{5.18}$$

The specific derivatives in these equations cannot be easily simplified. Due to the coupled material model and the complexity of the equations, the resulting derivation thereof is quite lengthy. Therefore, for readability reasons, further specifications are omitted here. The detailed breakdown can be found in Appendix B.1. Especially the derivatives regarding $\boldsymbol{C}^{\mathrm{p}-1}$ require special attention, since the local material model is defined in the principal space and therefore requires consideration of derivatives of the eigenvalues and eigenprojections. Furthermore, the history sensitivities are independent of the chosen design $\boldsymbol{s}$ and therefore applicable for any choice of design.

5.3.3 History sensitivity update

The internal variables generally change in value throughout the applied loads within a given simulation. Consequently, the variation of the history variables changes as well. Specifically, the history sensitivities $\boldsymbol{h}^{\boldsymbol{w}}$ in (5.15) -(5.18) require the total variation of the history variables $\delta\boldsymbol{h}_n$ from the previous time step t_n. Therefore, the total variation of internal history variables $\delta\boldsymbol{h}$ is necessary to be computed, cf. [146, 147, 233]. It reads

$$\begin{aligned}\delta\boldsymbol{h} &= \delta_{\boldsymbol{w}}\boldsymbol{h} + \delta_{\boldsymbol{s}}\boldsymbol{h} + \delta_{\boldsymbol{h}_n}\boldsymbol{h}\\ &= \frac{\partial\boldsymbol{h}}{\partial\boldsymbol{w}}\cdot\delta\boldsymbol{w} + \frac{\partial\boldsymbol{h}}{\partial\boldsymbol{s}}\cdot\delta\boldsymbol{s} + \frac{\partial\boldsymbol{h}}{\partial\boldsymbol{h}_n}\cdot\delta\boldsymbol{h}_n.\end{aligned} \tag{5.19}$$

This total variation has to be updated after each converged global load step, where a change in applied load leads to a change in the history variables $\boldsymbol{h}$. Therefore, special attention is required within the numerical framework, which is described in detail for the discretised quantities later on, see Sect. 5.7. Similar to the pseudo load, this total variation of the history field is design choice dependent and has to be resolved independently for each specific design variable $\boldsymbol{s}$ in their corresponding section.

Considering the above equation, it is useful here to derive the sensitivity operator $\tilde{\boldsymbol{s}}$, cf. [129, 145], by restructuring the initial total variation of the weak form equilibrium (5.12). At a current global equilibrium solution point $(\boldsymbol{w}^*,\hat{\boldsymbol{s}},\boldsymbol{h}_n^*)$, with fixed design $\hat{\boldsymbol{s}}$, this operator reads

$$\delta\boldsymbol{w} = \tilde{\boldsymbol{s}}(\boldsymbol{w}^*,\hat{\boldsymbol{s}},\boldsymbol{h}_n^*;\delta\boldsymbol{s})\cdot\delta\boldsymbol{s}, \tag{5.20}$$

The sensitivity operator $\tilde{\boldsymbol{s}}$ can be further specified for the mechanical and damage part, i.e.

$$\delta\boldsymbol{\varphi} =: \tilde{s}^{\varphi}(\boldsymbol{w}^*,\hat{\boldsymbol{s}},\boldsymbol{h}_n^*;\delta\boldsymbol{s})\cdot\delta\boldsymbol{s} \quad \text{and} \quad \delta\phi =: \tilde{s}^{\phi}(\boldsymbol{w}^*,\hat{\boldsymbol{s}},\boldsymbol{h}_n^*;\delta\boldsymbol{s})\cdot\delta\boldsymbol{s}, \tag{5.21}$$

respectively. Since this operator is directly dependent on the choice of design variable, they have to be derived separately. Furthermore, the direct calculation of these sensitivity operators is only possible after discretisation and are therefore presented in their respective sections.

This operator enables (5.19) to be rewritten, such that

$$\delta\boldsymbol{h} = \left[\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{w}}\circ\tilde{\boldsymbol{s}} + \frac{\partial\boldsymbol{h}}{\partial\boldsymbol{s}}\right]\cdot\delta\boldsymbol{s} + \frac{\partial\boldsymbol{h}}{\partial\boldsymbol{h}_n}\cdot\delta\boldsymbol{h}_n =: \boldsymbol{z}(\boldsymbol{w}^*,\hat{\boldsymbol{s}},\boldsymbol{h}_n^*;\delta\boldsymbol{s})\cdot\delta\boldsymbol{s}. \tag{5.22}$$

Therein, $\boldsymbol{z}$ is defined as history sensitivity operator. Since this equation requires the total variation of the history field form the previous time step t_n, i.e. $\delta\boldsymbol{h}_n = \boldsymbol{z}_n(\boldsymbol{w}^*,\hat{\boldsymbol{s}},\boldsymbol{h}_{n-1}^*;\delta\boldsymbol{s})$, this data has to be stored after each iteration to be made available for the next update, similar to the history variables themself. As a consequence, the total variation of the history field is reformulated, such that it only depends on the chosen design variables $\boldsymbol{s}$, i.e.

$$\delta \boldsymbol{h} = \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{w}} \circ \tilde{\boldsymbol{s}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{s}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_n} \circ \boldsymbol{z}_n\right] \cdot \delta \boldsymbol{s}. \tag{5.23}$$

For the first time step, the variation of the history field for $t = t_0$ is initially assumed to vanish, i.e. $\delta \boldsymbol{h}_0 = \boldsymbol{z}_0 = \boldsymbol{0}$, such that the initial update simplifies to

$$\delta \boldsymbol{h}_1 = \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{w}} \circ \tilde{\boldsymbol{s}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{s}}\right] \cdot \delta \boldsymbol{s}. \tag{5.24}$$

5.3.4 Geometric sensitivities

For general shape optimisation, the underlying geometry of the problem is considered as the design variables. In the context of the previously applied variational approach to the ductile damage model of Chap. 4, this means the referential coordinates $\boldsymbol{X}$. Their variation is thereby denoted as $\delta\boldsymbol{X}$. Note that, even though throughout this section this variation is denoted, actually the variation is reduced to the mapping $\boldsymbol{\kappa}$, see (5.8). However, for ease of reading the aforementioned denotation is chosen instead. Since the resulting gradients describe the sensitivities w.r.t. the geometry $\boldsymbol{X}$, they are also labelled as geometric sensitivities in this work. Hence, in this section, the previous variations for a not further specified design $\boldsymbol{s}$ are now specifically chosen to be $\boldsymbol{s} = \boldsymbol{X}$, i.e. the referential design of a given problem in application for shape optimisation.

Pseudo load. The pseudo load terms can thus be specified as

$$\begin{aligned} p_X^{\varphi} = \delta_{\boldsymbol{X}} r^{\varphi} = &\int_{\mathcal{B}_\zeta} \delta_{\boldsymbol{X}} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta \\ &- \int_{\mathcal{B}_\zeta} \boldsymbol{P}^{\mathrm{K}} : \delta_{\boldsymbol{X}} \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta \\ &+ \int_{\mathcal{B}_\zeta} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \, \delta_{\boldsymbol{X}} J_{\mathrm{K}} \, \mathrm{d}V_\zeta, \end{aligned} \tag{5.25}$$

for the mechanical part, and

$$\begin{aligned} p_X^{\phi} = \delta_{\boldsymbol{X}} r^{\phi} = &\int_{\mathcal{B}_\zeta} \delta_{\boldsymbol{X}} \boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^{\phi} \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta \\ &+ \int_{\mathcal{B}_\zeta} \boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^{\phi} \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta \\ &+ \int_{\mathcal{B}_\zeta} \delta_{\boldsymbol{X}} Y \, \eta^{\phi} \, J_{\mathrm{K}} \, \mathrm{d}V_\zeta \\ &+ \int_{\mathcal{B}_\zeta} \left[\boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^{\phi} + Y \, \eta^{\phi}\right] \delta_{\boldsymbol{X}} J_{\mathrm{K}} \, \mathrm{d}V_\zeta \end{aligned} \tag{5.26}$$

for the damage part. The enhanced kinematics, detailed in Sect. 5.2, are utilised here to separate the implicit dependencies of geometry and deformation. This allows reconstructing the partial variation of the gradient of the test functions as

$$\delta_{\boldsymbol{X}} \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} = -\nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \cdot \nabla_{\boldsymbol{X}} \delta \boldsymbol{X}, \tag{5.27}$$

$$\delta_{\boldsymbol{X}} \nabla_{\boldsymbol{X}} \eta^{\phi} = -\nabla_{\boldsymbol{X}} \eta^{\phi} \cdot \nabla_{\boldsymbol{X}} \delta \boldsymbol{X}, \tag{5.28}$$

cf. e.g. [17, 155], where the general relations within the enhanced kinematic concept are specified further. Therein, also the partial variation of the determinant of the geometry gradient is derived

$$\delta_{\boldsymbol{X}} J_{\mathrm{K}} = J_{\mathrm{K}} \nabla_{\boldsymbol{X}} \cdot \delta \boldsymbol{X}. \tag{5.29}$$

Using the information from (5.10) to compute the partial variation of the deformation gradient, the partial variations $\delta_{\boldsymbol{X}} \boldsymbol{P}^{\mathrm{K}}$ and $\delta_{\boldsymbol{X}} Y$ can be split, such that

$$\delta_{\boldsymbol{X}} \boldsymbol{P}^{\mathrm{K}} = \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \boldsymbol{F}} : \delta_{\boldsymbol{X}} \boldsymbol{F} \quad \text{and} \quad \delta_{\boldsymbol{X}} Y = \frac{\partial Y}{\partial \boldsymbol{F}} : \delta_{\boldsymbol{X}} \boldsymbol{F}, \tag{5.30}$$

where the arising derivatives are already known as the tangent contribution of the material model, see Sect. 4.6 and Fig. 4.4.

This finally results in

$$\begin{aligned} p_X^{\varphi} = & \int_{\mathcal{B}_0} \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} : \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \boldsymbol{F}} : \delta_{\boldsymbol{X}} \boldsymbol{F} \, \mathrm{d}V \\ & - \int_{\mathcal{B}_0} \boldsymbol{P}^{\mathrm{K}} : [\nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \cdot \nabla_{\boldsymbol{X}} \delta \boldsymbol{X}] \, \mathrm{d}V \\ & + \int_{\mathcal{B}_0} \boldsymbol{P}^{\mathrm{K}} : \nabla_{\boldsymbol{X}} \boldsymbol{\eta}^{\varphi} \, [\nabla_{\boldsymbol{X}} \cdot \delta \boldsymbol{X}] \, \mathrm{d}V, \end{aligned} \tag{5.31}$$

for the mechanical part, and

$$\begin{aligned} p_X^{\phi} = & -\int_{\mathcal{B}_0} \nabla_{\boldsymbol{X}} \eta^{\phi} \cdot [\boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \delta \boldsymbol{X}] \, \mathrm{d}V \\ & - \int_{\mathcal{B}_0} \boldsymbol{Y} \cdot \left[\nabla_{\boldsymbol{X}} \eta^{\phi} \cdot \nabla_{\boldsymbol{X}} \delta \boldsymbol{X}\right] \, \mathrm{d}V \\ & + \int_{\mathcal{B}_0} \eta^{\phi} \, \frac{\partial Y}{\partial \boldsymbol{F}} : \delta_{\boldsymbol{X}} \boldsymbol{F} \, \mathrm{d}V \\ & + \int_{\mathcal{B}_0} \left[\boldsymbol{Y} \cdot \nabla_{\boldsymbol{X}} \eta^{\phi} + Y \, \eta^{\phi}\right] \nabla_{\boldsymbol{X}} \cdot \delta \boldsymbol{X} \, \mathrm{d}V, \end{aligned} \tag{5.32}$$

for the damage part. In these equations, the integrals are mapped back to the referential configuration using relation (5.6).

History sensitivity update. The history sensitivity update of (5.19) here takes the specific form

$$\delta \boldsymbol{h} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{F}} : [\delta_{\boldsymbol{\varphi}} \boldsymbol{F} + \delta_{\boldsymbol{X}} \boldsymbol{F}] + \left[\frac{\partial \boldsymbol{h}}{\partial \phi} \circ \tilde{s}^{\phi}_{X} \right] \cdot \delta \boldsymbol{X} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_n} \cdot \delta \boldsymbol{h}_n, \tag{5.33}$$

where the partial variation of the deformation gradient have been specified in (5.10). Furthermore, the sensitivity operator from (5.21) with the specific design choice $\boldsymbol{s} = \boldsymbol{X}$ is required here.

Again, the partial derivatives in this equation are quite lengthy and a concise description is difficult. For readability reasons, the length formulae can be found in Appendix B.2.

5.4 Discrete ductile damage sensitivities

In order to implement the sensitivities into the numerical FE framework, the derived equations need to be discretised. Herein, the same discretisation as for the FE implementation of the material model are used, i.e. (4.70)-(5.34). While not specifically stated here, as it follows in the same fashion as for the discretisation of the weak forms in Sect. 4.6.1, the test functions $\delta \boldsymbol{\upvarphi}, \delta \upphi$ and $\delta \mathbf{X}$ have to be extracted from the resulting discretised integrals, as the equations have to hold even for arbitrary test values of these test functions. Only the resulting matrices and their assemblies are listed here. The missing discrete forms are still required, which read

$$\delta \boldsymbol{\varphi} \approx \delta \boldsymbol{\varphi}^h = \sum_{A=1}^{n^{\varphi}_{\text{en}}} \delta \boldsymbol{\upvarphi}_A N^{\varphi}_A \quad \text{and} \quad \phi \approx \phi^h = \sum_{B=1}^{n^{\phi}_{\text{en}}} \delta \upphi_B N^{\phi}_B, \tag{5.34}$$

for the variation of the field variables, and furthermore

$$\nabla_{\boldsymbol{X}} \delta \boldsymbol{\varphi} \approx \nabla_{\boldsymbol{X}} \delta \boldsymbol{\varphi}^h = \sum_{A=1}^{n^{\varphi}_{\text{en}}} \delta \boldsymbol{\upvarphi}_A \otimes \nabla_{\boldsymbol{X}} N^{\varphi}_A \quad \text{and} \quad \nabla_{\boldsymbol{X}} \delta \phi \approx \nabla_{\boldsymbol{X}} \delta \phi^h = \sum_{B=1}^{n^{\phi}_{\text{en}}} \delta \upphi_B \nabla_{\boldsymbol{X}} N^{\phi}_B, \tag{5.35}$$

for their gradients.

5.4.1 History sensitivity matrix

As stated in Sect. 5.3.2, the presented equations for the history sensitivity are valid for both choices of design utilised within this work, i.e. nodal coordinates $\mathbf{X}$ and later material parameters $\mathbf{m}_{\text{p}}$. Hence, their discretisation into the history sensitivity matrices will be presented here and read

$$\mathbf{H}^{\varphi}_{eA} = \int_{\mathcal{B}^e_0} [\mathbf{I} \otimes \nabla_{\boldsymbol{X}} N^{\varphi}_A] : \frac{\partial \mathbf{P}^{\text{K}}}{\partial \mathbf{h}_n} \, \mathrm{d}V \tag{5.36}$$

for the mechanical part, and

$$\mathbf{H}^{\phi}_{eB} = \int_{\mathcal{B}^e_0} N^{\phi}_B \frac{\partial Y}{\partial \mathbf{h}_n} \, \mathrm{d}V, \tag{5.37}$$

for the damage part. The history sensitivity matrix can be assembled in a standard fashion, i.e.

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{H}^{\varphi}_e \\ \mathbf{H}^{\phi}_e \end{bmatrix}, \quad \mathbf{H} = \mathop{\mathsf{A}}_{e}^{n_{\mathrm{el}}} \mathbf{H}_e, \quad \mathbf{H} \in \mathbb{R}^{n_{\mathrm{dof}} \times n_{\mathrm{hv}}}, \tag{5.38}$$

with n_{dof} as the number of degrees of freedom, see Sect. 4.6.1, and n_{hv} the number of history variables. The history matrix $\mathbf{H}$ is therefore not quadratic.

5.4.2 Discrete geometric sensitivities

Pseudo load matrix. By applying the discretisation to the continuous pseudo load terms in (5.31) and (5.32), in addition to extracting the test functions, one obtains

$$\begin{aligned} \mathbf{P}^{\varphi}_{X,eAC} = &- \int_{\mathcal{B}^e_0} [\mathbf{I} \otimes \nabla_{\boldsymbol{X}} N^{\varphi}_A] : \frac{\partial \mathbf{P}^{\mathrm{K}}}{\partial \mathbf{F}} : \left[\mathbf{G} \,\overline{\otimes}\, \nabla_{\boldsymbol{X}} N^X_C\right] \mathrm{d}V \\ &- \int_{\mathcal{B}^e_0} \mathbf{P}^{\mathrm{K}} \cdot \nabla_{\boldsymbol{X}} N^X_C \otimes \nabla_{\boldsymbol{X}} N^{\varphi}_A \, \mathrm{d}V \\ &+ \int_{\mathcal{B}^e_0} \mathbf{P}^{\mathrm{K}} \cdot \nabla_{\boldsymbol{X}} N^{\varphi}_A \otimes \nabla_{\boldsymbol{X}} N^X_C \, \mathrm{d}V, \end{aligned} \tag{5.39}$$

for the mechanical contribution, and

$$\begin{aligned} \mathbf{P}^{\phi}_{X,eBC} = &- \int_{\mathcal{B}^e_0} \mathbf{Y} \cdot \nabla_{\boldsymbol{X}} N^X_C \nabla_{\boldsymbol{X}} N^{\phi}_B \, \mathrm{d}V \\ &- \int_{\mathcal{B}^e_0} \nabla_{\boldsymbol{X}} N^{\phi}_B \cdot \nabla_{\boldsymbol{X}} N^X_C \mathbf{Y} \, \mathrm{d}V \\ &- \int_{\mathcal{B}^e_0} N^{\phi}_B \frac{\partial Y}{\partial \mathbf{F}} : \left[\mathbf{G} \,\overline{\otimes}\, \nabla_{\boldsymbol{X}} N^X_C\right] \mathrm{d}V \\ &+ \int_{\mathcal{B}^e_0} \left[\mathbf{Y} \cdot \nabla_{\boldsymbol{X}} N^{\phi}_B + Y N^{\phi}_B\right] \nabla_{\boldsymbol{X}} N^X_C \, \mathrm{d}V, \end{aligned} \tag{5.40}$$

for the nonlocal damage contribution, where (5.10) has been used for the partial variation of the deformation gradient and the matrix of the displacement gradient $\mathbf{G} = \nabla_{\boldsymbol{X}} \mathbf{u}$ is introduced. Note, that normally this quantity is denoted as $\mathbf{H}$. However, in this work $\mathbf{H}$ represents the history sensitivity matrix, hence the chosen variable to depict the displacement gradient.

Like the stiffness matrix, the pseudo load matrix has to be assembled in a similar manner, i.e.

$$\mathbf{P}_{X,e} = \begin{bmatrix} \mathbf{P}^{\varphi}_{X,e} \\ \mathbf{P}^{\phi}_{X,e} \end{bmatrix}, \quad \mathbf{P}_X = \mathop{\mathsf{A}}_{e}^{n_{\text{el}}} \mathbf{P}_{X,e}, \quad \mathbf{P}_X \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{X}}}. \tag{5.41}$$

In contrast to the stiffness matrix, the pseudo load matrix $\mathbf{P}_X$ is not quadratic, i.e. $n_{\text{dof}} \neq n_{\text{X}}$. Here, $n_{\text{X}} = n^{\varphi}_{\text{dim}} \cdot n_{\text{np}} = n^{X}_{\text{dim}} \cdot n_{\text{np}}$ encompasses the geometrical three-dimensional space of the FE simulation. With the addition of the nonlocal damage variable ϕ, the number of degrees of freedom at each node is larger by one, i.e. $n_{\text{dof}} = (n^{\varphi}_{\text{dim}}+1)\cdot n_{\text{np}}$. Hence, $n_{\text{dof}} > n_{\text{X}}$, leading to the non-quadratic nature of the pseudo load matrix.

Sensitivity matrix. The variation of the history variables $\delta\mathbf{h}_n$, see (5.11), is required in its discretised form. However, its computation requires the sensitivity matrix $\mathbf{S}$, which is derived in the following, see (5.47). Therefore, the total variation of the history field is derived afterwards, see (5.49), and the resulting sub matrices $\mathbf{Z}^{\bullet}$ assumed to be known here. Their values are stored in the matrix $\mathbf{Z}_{X,e}$, i.e.

$$\mathbf{Z}_{X,e} = \begin{bmatrix} \mathbf{Z}^{\boldsymbol{C}}_{X,e} \\ \mathbf{Z}^{\alpha}_{X,e} \\ \mathbf{Z}^{d}_{X,e} \end{bmatrix} \in \mathbb{R}^{n_{\text{hv}} \times n_{\text{X}}}, \tag{5.42}$$

with the sub-matrices

$$\mathbf{Z}^{\boldsymbol{C}}_{X,e} = \left[\frac{\partial \mathbf{C}^{\text{p}-1}_{n}}{\partial \mathbf{X}}\right] \in \mathbb{R}^{9 \times n_{\text{X}}}, \quad \mathbf{Z}^{\alpha}_{X,e} = \left[\frac{\partial \alpha_n}{\partial \mathbf{X}}\right] \in \mathbb{R}^{1 \times n_{\text{X}}}, \quad \mathbf{Z}^{d}_{X,e} = \left[\frac{\partial d_n}{\partial \mathbf{X}}\right] \in \mathbb{R}^{1 \times n_{\text{X}}}. \tag{5.43}$$

Note, that the matrix $\mathbf{Z}_{X,e}$ is not assembled in a standard manner. The reasoning is elaborated on in Sect. 5.7 and not further detailed here.

Combining the history sensitivity matrix $\mathbf{H}_e$ as well as the discretised total variation of the history field $\mathbf{Z}_{X,e}$, according to (5.15) and (5.16), allows their combination into the elemental matrix $\mathbf{Q}_{X,e}$. This new quantity connects the structural response with the design changes that affect the history variables. The element-wise contributions for the mechanical and damage part read

$$\mathbf{Q}^{\varphi}_{X,eAC} = \mathbf{H}^{\varphi}_{eA} \cdot \mathbf{Z}_{X,eC} \quad \text{and} \quad \mathbf{Q}^{\phi}_{X,eBC} = \mathbf{H}^{\phi}_{eB} \cdot \mathbf{Z}_{X,eC}, \tag{5.44}$$

respectively, which are globally assembly in a standard fashion, i.e.

$$\mathbf{Q}_{X,e} = \begin{bmatrix} \mathbf{Q}^{\varphi}_{X,e} \\ \mathbf{Q}^{\phi}_{X,e} \end{bmatrix}, \quad \mathbf{Q}_e = \mathop{\mathsf{A}}_{e}^{n_{\text{el}}} \mathbf{Q}_{X,e}, \quad \mathbf{Q}_e \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{hv}}}. \tag{5.45}$$

Finally, this allows the discrete matrix representation of (5.12), i.e.

$$\delta\mathbf{r} = \mathbf{K} \cdot \delta\mathbf{w} + \mathbf{P}_X \cdot \delta\mathbf{X} + \mathbf{H} \cdot \delta\mathbf{h}_n = \mathbf{0}. \tag{5.46}$$

Reordering of this equation, and using (5.45), leads to

$$\delta\mathbf{w} = -\mathbf{K}^{-1} \cdot [\mathbf{P}_X + \mathbf{Q}_X] \cdot \delta\mathbf{X} =: \mathbf{S}_X \cdot \delta\mathbf{X}, \tag{5.47}$$

allowing the definition of the global sensitivity matrix $\mathbf{S}_X \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{X}}}$. This is the discretised form of the sensitivity operator $\tilde{\boldsymbol{s}}_X$, see (5.20). This quantity defines how the underlying system $\delta\mathbf{w}$ reacts to a change in design $\delta\mathbf{X}$, and is the foundation for the derivation of gradients for the objectives and constraints within the optimisation problems described in Sect. 5.6. Similarly to all previous quantities, the sensitivity matrix can be decomposed into a mechanical and damage part as well, i.e.

$$\mathbf{S}_X = \begin{bmatrix} \mathbf{S}_X^{\varphi} \\ \mathbf{S}_X^{\phi} \end{bmatrix}. \tag{5.48}$$

Discrete history sensitivity update. In the previous equations, the discrete matrix description for the total history variation (5.19) is utilised. Since the history variables are stored for each Gauss point on the element level, the matrices have to be computed and stored at those levels as well, and afterwards assembled into their corresponding history sensitivity contribution. The discrete total history variation, by utilising (5.48), thus reads

$$\begin{aligned} \delta\mathbf{h} = {} & \left[\frac{\partial \mathbf{h}}{\partial \mathbf{F}} \cdot \nabla_{\mathbf{X}} N_A^{\varphi} \cdot \mathbf{S}_{X,eC}^{\varphi} + N_B^{\phi} \frac{\partial \mathbf{h}}{\partial \boldsymbol{\phi}} \cdot \mathbf{S}_{X,eC}^{\phi} - \frac{\partial \mathbf{h}}{\partial \mathbf{F}} : \left[\mathbf{G} \,\overline{\otimes}\, \nabla_{\mathbf{X}} N_C^X \right] + \frac{\partial \mathbf{h}}{\partial \mathbf{h}_n} \cdot \mathbf{Z}_{n,eC} \right] \cdot \delta\mathbf{X} \\ =: {} & \mathbf{Z}_{X,eC} \cdot \delta\mathbf{X}. \end{aligned} \tag{5.49}$$

Additionally, as was already mentioned in Sect. 5.3.3, for the first time step $t = t_0$, one can assume $\mathbf{h}_0 = \delta\mathbf{h}_0 = \mathbf{0}$, hence $\mathbf{Z}_0 = \mathbf{0}$ and $\mathbf{Q}_1 = \mathbf{0}$, which leads to the simplified computation of the sensitivity matrix

$$\mathbf{S}_1 = -\mathbf{K}_1^{-1} \cdot \mathbf{P}_1, \tag{5.50}$$

for the first pseudo-time increment at $t = t_1$.

Remark 5.1 The split into unknown a and known degrees of freedom b, see (4.88), can also be applied to the design variables $\mathbf{s}$, the pseudo load matrix $\mathbf{P}$ and the matrix $\mathbf{Q}$, i.e.

$$\delta\mathbf{s} = \begin{bmatrix} \mathbf{s}_a \\ \mathbf{s}_b \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{aa} & \mathbf{P}_{ab} \\ \mathbf{P}_{ba} & \mathbf{P}_{bb} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{aa} & \mathbf{Q}_{ab} \\ \mathbf{Q}_{ba} & \mathbf{Q}_{bb} \end{bmatrix}. \tag{5.51}$$

Since the prescribed displacements $\mathbf{w}_b$ are known and hence their variation vanishes, i.e. $\delta\mathbf{w}_b = \mathbf{0}$, one has to solve the sensitivity matrix for the free, unknown contributions a. The reduced sensitivity matrix reads

$$\mathbf{S}_a = -\mathbf{K}_{aa}^{-1} \cdot [\mathbf{P}_a + \mathbf{Q}_a], \tag{5.52}$$

where $\bullet_a = [\bullet_{aa} \quad \bullet_{ab}]$, resulting in the final split for the sensitivity matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_a \\ \mathbf{S}_b \end{bmatrix} = \begin{bmatrix} \mathbf{S}_a \\ \mathbf{0} \end{bmatrix} . \tag{5.53}$$

5.5 Discrete material sensitivities

In addition to the geometric sensitivities, the ductile damage material model of Chap. 4 is also enhanced for gradient information w.r.t. the constitutive material parameters $\mathbf{m}_\mathrm{p}$. In contrast to the geometric description $\boldsymbol{X}$, i.e. the continuous description $\boldsymbol{X} = \kappa(\boldsymbol{\Theta})$ from the intrinsic viewpoint of the enhanced kinematics, see (5.8), the constitutive material parameters are inherently discrete quantities. Therefore, a continuous derivation of the sensitivities akin to the geometric sensitivities is not possible. Instead, the discrete functions of the weak form have to be utilised. Nonetheless, these discrete quantities for the material design $\mathbf{s} = \mathbf{m}_\mathrm{p}$ follow a very similar course of derivation compared to the geometric problem in the previous section.

Due to the complexity of the material model, there are many parameters to consider and they are summarised here again. The parameters K and G are the bulk and shear modulus, respectively, and govern the elastic behaviour of the material. The yield stress $\sigma_{\mathrm{y}0}$, the hardening modulus h and the hardening exponent n_p control the plastic behaviour. To control the damage behaviour many more parameters are needed. These encompass the damage threshold q_min and damage exponent n_d, the damage rate factor η, the volumetric and isochoric damage factors ξ_vol and ξ_iso, respectively, the threshold factor ξ_q and effective stress factor ξ_m, as well as the coupling factor η_α. Finally, the numerical regularisation is governed by the penalty factor β_d and the regularisation parameter c_d. While theoretically, the referential density ρ_0 could be included, body forces are omitted within this work. The list of design variables thereby consists of 15 parameters and reads

$$\mathbf{m}_\mathrm{p} := [E,\nu;\, \sigma_{\mathrm{y}0},h,n_\mathrm{p};\, \eta,\xi_\mathrm{vol},\xi_\mathrm{iso},\xi_\mathrm{q},\xi_\mathrm{m},\eta_\alpha,q_\mathrm{min},n_\mathrm{d};\, c_\mathrm{d},\beta_\mathrm{d}] \, . \tag{5.54}$$

Since the resulting gradients describe the sensitivities w.r.t. the material behaviour $\mathbf{m}_\mathrm{p}$, they are also labelled as material sensitivities in this work.

Pseudo load matrix. Since the constitutive material parameters only have a contribution within the model description, no implicit dependencies within the integrals of the weak form arise. Therefore, they reduce in complexity, as only the nonlocal driving forces are implicitly dependent on this set of design variables. The discrete pseudo load matrix for the constitutive parameters therefore reads

$$\mathbf{P}^{\varphi}_{m,eAC} = \int_{\mathcal{B}_0^e} [\mathbf{I} \otimes \nabla_{\boldsymbol{X}} N^{\varphi}_A] : \frac{\partial \mathbf{P}^\mathrm{K}}{\partial \mathbf{m}_\mathrm{p}} \, \mathrm{d}V, \tag{5.55}$$

for the mechanical contribution, and

$$\begin{aligned}\mathbf{P}^{\phi}_{m,eBC} = &- \int_{\mathcal{B}_0^e} \nabla_{\mathbf{X}} N^{\phi}_B \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{m}_{\mathrm{p}}} \,\mathrm{d}V \\ &+ \int_{\mathcal{B}_0^e} N^{\phi}_B \frac{\partial Y}{\partial \mathbf{m}_{\mathrm{p}}} \,\mathrm{d}V,\end{aligned} \tag{5.56}$$

for the nonlocal damage contribution. The assembly is achieved in a standard fashion, i.e.

$$\mathbf{P}_{m,e} = \begin{bmatrix} \mathbf{P}^{\varphi}_{m,e} \\ \mathbf{P}^{\phi}_{m,e} \end{bmatrix}, \quad \mathbf{P}_m = \mathop{\mathsf{A}}_{e}^{n_{\mathrm{el}}} \mathbf{P}_{m,e}, \quad \mathbf{P}_m \in \mathbb{R}^{n_{\mathrm{dof}} \times n_{\mathrm{m}}}. \tag{5.57}$$

Again, the discretised pseudo load matrix $\mathbf{P}_m$ is not quadratic. However, in contrast to the geometric setting, the dimension n_{m} is independent of the chosen mesh and instead material model dependent. For the applied model here, $n_{\mathrm{m}} = 15$, since the material model utilises 15 unique constitutive material parameters.

While the resulting integrals are significantly shorter, compared to the geometrical case, the remaining derivatives cannot be resolved into already known quantities, e.g. quantities derived for the stiffness contribution, and require additional computation. They are again not specifically derived here and rather concisely listed in Appendix B.3. Their derivation is separated into elastic, plastic, damage and regularisation parts. An exception in this regard within the above integral (5.56), is the partial derivative of the tensorial nonlocal driving force $\mathbf{Y}$. Since this quantity only depends on the regularisation parameter c_{d}, the partial derivative is very easy to compute, i.e.

$$\frac{\partial \mathbf{Y}}{\partial \mathbf{m}_{\mathrm{p},i}} = \begin{cases} \nabla_{\mathbf{X}} \phi & \mathbf{m}_{\mathrm{p},i} = c_{\mathrm{d}} \\ 0 & \text{else.} \end{cases} \tag{5.58}$$

Sensitivity matrix. Following the same methodology as in the geometrical case, the history sensitivity update in its discrete form reads

$$\mathbf{Z}_{m,e} = \begin{bmatrix} \mathbf{Z}^{\boldsymbol{C}}_{m,e} \\ \mathbf{Z}^{\alpha}_{m,e} \\ \mathbf{Z}^{d}_{m,e} \end{bmatrix} \in \mathbb{R}^{n_{\mathrm{hv}} \times n_{\mathrm{m}}}, \tag{5.59}$$

with the sub-matrices

$$\begin{aligned} &\mathbf{Z}^{\boldsymbol{C}}_{m,e} = \left[\frac{\partial \mathbf{C}^{\mathrm{p}-1}_n}{\partial \mathbf{m}_{\mathrm{p}}} \right] \in \mathbb{R}^{9 \times n_{\mathrm{m}}}, \quad \mathbf{Z}^{\alpha}_{m,e} = \left[\frac{\partial \alpha_n}{\partial \mathbf{m}_{\mathrm{p}}} \right] \in \mathbb{R}^{1 \times n_{\mathrm{m}}}, \\ &\mathbf{Z}^{d}_{m,e} = \left[\frac{\partial d_n}{\partial \mathbf{m}_{\mathrm{p}}} \right] \in \mathbb{R}^{1 \times n_{\mathrm{m}}}. \end{aligned} \tag{5.60}$$

and therefore the matrix $\mathbf{Q}_{m,e}$, which combines the history sensitivity update with the history sensitivity matrix $\mathbf{H}_e$. Since the latter quantity is design independent, as was highlighted during its derivation, it can find application without further adjustments here. Again, a split of mechanical and damage part is made, such that

$$\mathbf{Q}^{\varphi}_{m,eA} = \mathbf{H}^{\varphi}_{eA} \cdot \mathbf{Z}_{m,e} \quad \text{and} \quad \mathbf{Q}^{\phi}_{m,eB} = \mathbf{H}^{\phi}_{eB} \cdot \mathbf{Z}_{m,e}, \tag{5.61}$$

respectively. Both quantities are globally assembly in a standard fashion, i.e.

$$\mathbf{Q}_{m,e} = \begin{bmatrix} \mathbf{Q}^{\varphi}_{m,e} \\ \mathbf{Q}^{\phi}_{m,e} \end{bmatrix}, \quad \mathbf{Q}_m = \overset{n_{\text{el}}}{\underset{e}{\mathsf{A}}} \mathbf{Q}_{m,e}, \quad \mathbf{Q}_m \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{m}}}. \tag{5.62}$$

This Finally allows the derivation for the sensitivity matrix $\mathbf{S}_m$ for the material sensitivities following the same routine as in the previous section, i.e.

$$\delta \mathbf{w} = -\mathbf{K}^{-1} \cdot [\mathbf{P}_m + \mathbf{Q}_m] \cdot \delta \mathbf{m}_{\text{p}} =: \mathbf{S}_m \cdot \delta \mathbf{m}_{\text{p}}, \tag{5.63}$$

which herein has dimension $\mathbf{S}_m \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{m}}}$. It describes the change in structural response $\delta \mathbf{w}$ of a system due to a change of the underlying constitutive material parameters $\delta \mathbf{m}_{\text{p}}$.

History sensitivity update. The discrete form of the history sensitivity updates for the constitutive material parameters reads

$$\begin{aligned} \delta \mathbf{h} = &\left[\frac{\partial \mathbf{h}}{\partial \mathbf{F}} \cdot \nabla_{\mathbf{X}} N^{\varphi}_A \cdot \mathbf{S}^{\varphi}_{m,e} + N^{\phi}_B \frac{\partial \mathbf{h}}{\partial \phi} \cdot \mathbf{S}^{\phi}_{m,e} + \frac{\partial \mathbf{h}}{\partial \mathbf{m}_{\text{p}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{h}_n} \cdot \mathbf{Z}_{n,e} \right] \cdot \delta \mathbf{m}_{\text{p}} \\ &=: \mathbf{Z}_{m,e} \cdot \delta \mathbf{m}_{\text{p}}. \end{aligned} \tag{5.64}$$

Similar to the geometric case, the submatrices $\mathbf{S}^{\varphi}_{m,e}$ and $\mathbf{S}^{\phi}_{m,e}$ are required for the history update of the material sensitivities. However, since the partial derivatives cannot be formulated into derivatives w.r.t. the deformation gradient, this history update requires both submatrices. Furthermore, the first and last partial derivative, i.e. the partial derivatives w.r.t. $\mathbf{F}$ and $\mathbf{h}_n$, respectively, are already known as they are required for the geometrical case as well, see (5.33). New in this regard, are the partial derivatives w.r.t. the constitutive parameters. Taking information from the pseudo load data in Appendix B.3 into account, i.e. the derivatives for the Lagrangian multipliers of the local Newton-Raphson, these remaining derivatives can be easily summarised and therefore are listed here. They read

$$\frac{\partial \mathbf{C}^{\text{p}-1}}{\partial \mathbf{m}_{\text{p}}} = -\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\text{p}}}{\partial \mathbf{m}_{\text{p}}}, \tag{5.65}$$

$$\frac{\partial \alpha}{\partial \mathbf{m}_{\text{p}}} = \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\text{p}}}{\mathrm{d}\mathbf{m}_{\text{p}}}, \tag{5.66}$$

$$\frac{\partial d}{\partial \mathbf{m}_{\text{p}}} = \frac{\mathrm{d}\Delta\lambda^{\text{d}}}{\mathrm{d}\mathbf{m}_{\text{p}}} \hat{f}^{\alpha} + \Delta\lambda^{\text{d}} \frac{\partial \hat{f}^{\alpha}}{\partial \mathbf{m}_{\text{p}}} + \Delta\lambda^{\text{d}} \left[\hat{f}^{\alpha} \right]' \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\text{p}}}{\mathrm{d}\mathbf{m}_{\text{p}}} \tag{5.67}$$

with

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \mathbf{m}_{\mathrm{p}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\mathbf{m}_{\mathrm{p}}} \hat{f}^{\boldsymbol{m}} \otimes \bar{\mathbf{v}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}} \frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \mathbf{m}_{\mathrm{p}}} \otimes \bar{\mathbf{v}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}} \left[\hat{f}^{\boldsymbol{m}}\right]' \frac{\partial d}{\partial \mathbf{m}_{\mathrm{p}}} \otimes \bar{\mathbf{v}}^{\mathrm{tr}}, \tag{5.68}$$

$$\frac{\partial \hat{f}^{\alpha}}{\partial \mathbf{m}_{\mathrm{p},i}} = \begin{cases} \alpha\, f^{\alpha} \frac{1+\epsilon}{[f^{\alpha}+\epsilon]^2} & \mathbf{m}_{\mathrm{p},i} = \eta_{\alpha} \\ 0 & \text{else,} \end{cases} \tag{5.69}$$

$$\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \mathbf{m}_{\mathrm{p},i}} = \begin{cases} \xi_{\mathrm{m}}\, d\, f^{\boldsymbol{m}} \frac{1+\epsilon}{[f^{\boldsymbol{m}}+\epsilon]^2} & \mathbf{m}_{\mathrm{p},i} = \eta \\ \eta\, d\, f^{\boldsymbol{m}} \frac{1+\epsilon}{[f^{\boldsymbol{m}}+\epsilon]^2} & \mathbf{m}_{\mathrm{p},i} = \xi_{\mathrm{m}} \\ 0 & \text{else.} \end{cases} \tag{5.70}$$

5.6 Objectives and constraints

For the optimisation problems in Chap. 6, their necessary objective functions and respective gradients have to be specified. Since different types of objective and constraints are considered for the geometrical and material sensitivities, respectively, their definitions and required sensitivities are presented in separate subsections. Note, that the quantity $\mathfrak{J}$ here can either take the form of a function, i.e. the objective function of a minimisation problem (3.1) or (3.2), or a functional which is discretised to be the objective function of said optimisation problems.

5.6.1 Shape optimisation

In a first example, the damage accumulation within a geometry is directly influenced to govern the optimisation problem. As such, the damage variables $\boldsymbol{\phi}_i$ at each elemental node n_{np}^{ϕ} are directly utilised. To capture their values in a scalar objective function $\mathfrak{J}$, the squared vector-norm of the damage vector $\boldsymbol{\phi}$ is chosen, i.e.

$$\mathfrak{J}^{\mathrm{d}} = \|\boldsymbol{\phi}(\mathbf{s})\|^2 = \sum_{i=1}^{n_{\mathrm{np}}^{\phi}} \boldsymbol{\phi}_i(\mathbf{X})^2, \tag{5.71}$$

and

$$\delta\mathfrak{J}^{\mathrm{d}} = \frac{\partial \|\boldsymbol{\phi}(\mathbf{s})\|^2}{\partial \mathbf{X}} \cdot \delta\mathbf{X}. \tag{5.72}$$

The gradients of this objective are straightforward to compute. Since the nodal damage quantities $\boldsymbol{\phi}_i$ are part of the set of field variables $\mathbf{w}$, their respective gradients consist of the subset of gradients within the sensitivity matrix $\mathbf{S}_X$, which in this case is the damage part $\mathbf{S}_X^{\phi}$. Application of the chain rule for the norm of the damage vector thus leads to

$$\frac{\partial \|\boldsymbol{\phi}(\mathbf{X})\|^2}{\partial \mathbf{X}} = 2\, \boldsymbol{\phi}(\mathbf{X}) \cdot \mathbf{S}_X^{\phi}. \tag{5.73}$$

Additionally, the nonlocal damage values $\boldsymbol{\phi}_i$ at each FE node $i = 1, \ldots, n_{\mathrm{np}}^{\varphi}$ can be chosen as constraints within the optimisation. In this case, the norm of the vector is not required, as each nodal damage quantity $\mathsf{\phi}_i(\mathbf{X})$ leads to an additional constraint, i.e.

$$\mathsf{\phi}_i(\mathbf{X}) \le \phi_{\text{crit}}, \quad i = 1, \ldots, n^{\phi}_{\text{np}}. \tag{5.74}$$

The critical or maximum prescribed damage value ϕ_{crit} is chosen initially and depends on the stated problem and chosen set of material parameters. The necessary gradients are again stored in the damage contribution $\mathbf{S}^{\phi}_{X}$ of the sensitivity matrix.

For the second optimisation example, the compliance C is chosen as the objective function. The aim of an optimisation problem with the compliance as the objective function is to generate a structure, which behaves stiffer under the applied boundary conditions. Maximisation of the stiffness results in a higher strain energy potential which the body under consideration can capture for an applied load. Rewriting this into a minimisation problem thus leads to a minimisation of the negative strain energy potential. The continuous functional which describes this quantity reads

$$\begin{aligned} \mathfrak{J}^{\text{C}} &= C(\boldsymbol{X}) = -\Pi^{\text{loc,el}}(\boldsymbol{X}) \\ &= -\int_{\mathcal{B}_0} \Psi^{\text{vol}}(\boldsymbol{X}) + \Psi^{\text{iso}}(\boldsymbol{X}) \, \text{d}V. \end{aligned} \tag{5.75}$$

This quantity only takes the elastic part $\Psi^{\text{loc,el}}$ of the local energy potential into account, since the nonlocal part Ψ^{nl} is only required for numerical reasons and cannot be interpreted physically. Additionally, external contributions for this potential are neglected due to the choice of boundary conditions in the example.

The gradients can be derived in the same fashion as described in the previous sections. The total variation, by already incorporating $\boldsymbol{s}_X$ and $\boldsymbol{z}_{X,n}$ for the corresponding variations of $\delta\boldsymbol{w}$ and $\delta\boldsymbol{h}_n$, respectively, therefore reads

$$\delta C = \left[\frac{\partial C}{\partial \boldsymbol{w}} \circ \boldsymbol{s}_X + \frac{\partial C}{\partial \boldsymbol{X}} + \frac{\partial C}{\partial \boldsymbol{h}_n} \circ \boldsymbol{z}_{X,n} \right] \cdot \delta\boldsymbol{X}. \tag{5.76}$$

Application of the discretisation (4.70)-(5.34) for the total sum of the bracket above leads to

$$\begin{aligned} \left[\frac{\partial C}{\partial \mathbf{X}} \right]_{eC} = &- \int_{\mathcal{B}_0^e} \left[\frac{\partial \Psi^{\text{vol}}}{\partial \mathbf{F}} + \frac{\partial \Psi^{\text{iso}}}{\partial \mathbf{F}} \right] \cdot \nabla_{\boldsymbol{X}} N^{\varphi}_C \cdot \mathbf{S}^{\varphi}_{X,eC} \, \text{d}V \\ &- \int_{\mathcal{B}_0^e} N^{\phi}_C \left[\frac{\partial \Psi^{\text{vol}}}{\partial \mathsf{\phi}} + \frac{\partial \Psi^{\text{iso}}}{\partial \mathsf{\phi}} \right] \cdot \mathbf{S}^{\phi}_{X,eC} \, \text{d}V \\ &+ \int_{\mathcal{B}_0^e} \left[\frac{\partial \Psi^{\text{vol}}}{\partial \mathbf{F}} + \frac{\partial \Psi^{\text{iso}}}{\partial \mathbf{F}} \right] : \left[\mathbf{G} \,\overline{\otimes}\, \nabla_{\boldsymbol{X}} N^X_C \right] \text{d}V \\ &- \int_{\mathcal{B}_0^e} \left[\frac{\partial \Psi^{\text{vol}}}{\partial \mathbf{h}_n} + \frac{\partial \Psi^{\text{iso}}}{\partial \mathbf{h}_n} \right] \cdot \mathbf{Z}_{X,n,eC} \, \text{d}V \\ &- \int_{\mathcal{B}_0^e} \left[\Psi^{\text{vol}} + \Psi^{\text{iso}} \right] \nabla_{\boldsymbol{X}} N^X_C \, \text{d}V, \end{aligned} \tag{5.77}$$

which has to be assembled in accordance to the previously described quantities. The specific partial derivatives within the integrals are derived in Appendix B.4.

Finally, all optimisation problems are subjected to geometric constraints regarding the volume V of the body under consideration. This additional constraint allows for better comparison since no material is added or subtracted to the geometries. The volume is computed is the sum of all elemental volumes and as such

$$V(\mathbf{X}) = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\mathcal{B}_0^e} \mathrm{d}V, \tag{5.78}$$

with its straightforward derivative

$$\left[\frac{\partial V}{\partial \mathbf{X}}\right]_{eC} = \int_{\mathcal{B}_0^e} \nabla_{\mathbf{X}} N_C^X \,\mathrm{d}V, \tag{5.79}$$

and assembly in the known fashion.

5.6.2 Reaction force sensitivities

The material parameter sensitivities are considered within the environment of parameter identification. This field of research aims to utilise experimental data $\bullet^{\mathrm{exp}}$ and tries to optimise simulation data $\bullet^{\mathrm{sim}}$, such that they match as closely as possible. Generally, the constitutive material parameters $\mathbf{m}_{\mathrm{p}}$ of the simulation model are chosen as the design variables, which directly influence the underlying structural response. A multi objective function considered in this setting has the following form

$$\mathfrak{J}^{\mathrm{PI}} = w^{\mathrm{F}} \mathfrak{J}^{\mathrm{F}} \left(F^{\mathrm{sim}}(t), F^{\mathrm{exp}}(t)\right) + w^{\mathrm{u}} \mathfrak{J}^{\mathrm{u}} \left(\mathit{\Delta}\mathbf{u}^{\mathrm{sim}}(\mathbf{X},t), \mathit{\Delta}\mathbf{u}^{\mathrm{exp}}(\mathbf{X},t)\right), \tag{5.80}$$

with the separate objectives

$$\mathfrak{J}^{\mathrm{F}} = \sum_{t=1}^{T} w_t \left[F_t^{\mathrm{sim}} - F_t^{\mathrm{exp}}\right]^2, \tag{5.81}$$

and

$$\mathfrak{J}^{\mathrm{u}} = \sum_{t=1}^{T} \frac{w_t}{n_{\mathrm{np}}} \sum_{i=1}^{n_{\mathrm{np}}} \|\mathit{\Delta}\mathbf{u}_i^{\mathrm{sim}} - \mathit{\Delta}\mathbf{u}_i^{\mathrm{exp}}\|_t^2, \tag{5.82}$$

see e.g. [190, 192, 215]. The first objective $\mathfrak{J}^{\mathrm{F}}$ compares simulated reaction forces with experimentally captured forces. The second objective $\mathfrak{J}^{\mathrm{u}}$ compares the simulated displacements with experimentally captured displacements, which can be achieved by means of digital image correlation (DIC). Instead of utilising the displacements $\mathbf{u}$, the change in displacement $\mathit{\Delta}\mathbf{u}$ is considered instead. With this method, each investigated points correlates their displacement with neighbouring points. The benefits of this approach are touched upon more in the cited literature above and not explicitly discussed here, since this measurement is not used in the presented work. The variables w_t allow further weighting of specific points within the experiment or simulation, respectively, while w^{F} and w^{u} enable weighting the two separate objectives for their contribution to $\mathfrak{J}^{\mathrm{PI}}$. Mainly this has to be considered to adjust their dimensions, as the two objectives $\mathfrak{J}^{\mathrm{F}}$ and $\mathfrak{J}^{\mathrm{u}}$ are generally not of same dimension. In this work however, only the first objective $\mathfrak{J}^{\mathrm{F}}$ is considered since either a) no DIC data for the experiments is available or b) it is not considered in the specific analysis.

To derive the sensitivities $\delta\mathfrak{J}^{\mathrm{F}}$, wherein only the forces from the simulation depend on the design, i.e. $\mathfrak{J}^{\mathrm{F}}\left(F^{\mathrm{sim}}(\mathbf{m}_{\mathrm{p}},t),F^{\mathrm{exp}}(t)\right)$, the total variation of the objective reads

$$\delta\mathfrak{J}^{\mathrm{F}} = \sum_{t=1}^{T} 2\,w_t\left[F_t^{\mathrm{sim}}(\mathbf{m}_{\mathrm{p}}) - F_t^{\mathrm{exp}}\right]\delta F_t^{\mathrm{sim}}, \tag{5.83}$$

with the total variation of the simulated reaction forces

$$\delta F_t^{\mathrm{sim}} = \delta_{\boldsymbol{w}} F_t^{\mathrm{sim}} + \delta_{\boldsymbol{m}_{\mathrm{p}}} F_t^{\mathrm{sim}} + \delta_{\boldsymbol{h}_n} F_t^{\mathrm{sim}} \tag{5.84}$$

A starting point for the derivation of this variation, recalling (4.75) and (4.88), is the linearised global equilibrium condition, split into internal and external part for unknown a and prescribed b contributions, i.e.

$$\mathbf{r} = \begin{bmatrix}\mathbf{r}_a^{\mathrm{int}}\\ \mathbf{r}_b^{\mathrm{int}}\end{bmatrix} - \begin{bmatrix}\mathbf{r}_a^{\mathrm{ext}}\\ \mathbf{r}_b^{\mathrm{ext}}\end{bmatrix} = \mathbf{0}. \tag{5.85}$$

The reaction forces F^{sim} considered in (5.81) for the objective $\mathfrak{J}^{\mathrm{F}}$ are computed at the prescribed boundary b and part of the external residual contribution $\mathbf{r}_b^{\mathrm{ext}}$. Hence, combination of (5.12) and (5.85) allows the derivation of the variation, such that

$$\begin{aligned}\delta\mathbf{r}_b^{\mathrm{ext}} &= \frac{\partial\mathbf{r}_b^{\mathrm{int}}}{\partial\mathbf{w}}\cdot\delta\mathbf{w} + \frac{\partial\mathbf{r}_b^{\mathrm{int}}}{\partial\mathbf{m}_{\mathrm{p}}}\cdot\delta\mathbf{m}_{\mathrm{p}} + \frac{\partial\mathbf{r}_b^{\mathrm{int}}}{\partial\mathbf{h}_n}\cdot\delta\mathbf{h}_n\\ &= \left[\mathbf{K}_{ba}\cdot\mathbf{S}_{m,a} + \mathbf{P}_{m,b} + \mathbf{H}_b\cdot\,\mathbf{Z}_{m,n}\right]\cdot\delta\mathbf{m}_{\mathrm{p}}.\end{aligned} \tag{5.86}$$

The simulation forces F^{sim} within the objective at each evaluated time step t are scalar valued. Since the forces from the discretised problem are stored at each nodal value, the scalar valued simulated force quantity F_t^{sim} and its variation $\delta F_t^{\mathrm{sim}}$ for each evaluated time step t can be captured by sum of all considered nodes, i.e.

$$F_t^{\mathrm{sim}} = \sum_{i=1}^{n_{\mathrm{fn}}}\mathbf{r}_{b,i}^{\mathrm{ext}} \quad\text{and}\quad \delta F_t^{\mathrm{sim}} = \sum_{i=1}^{n_{\mathrm{fn}}}\delta\mathbf{r}_{b,i}^{\mathrm{ext}}\,, \tag{5.87}$$

where n_{fn} are the number of nodes where the forces are taken into account. Generally, the simulated time steps t^{sim} and the experimental time steps t^{exp} do not coincide. Therefore either result has to be interpolated. In this work, the compared data $\bullet^{\mathrm{exp}}$ is interpolated, since the time steps within each simulation might vary due to the applied automatic time step control.

Note, that the objective $\mathfrak{J}^{\mathrm{F}}$ takes the complete loading history $t \in [0,T]$ into consideration. Since the sensitivity analysis for each time load step t is conducted after each converged global Newton iteration, the resulting iterative objective and gradient have to be stored for post-processing. After the final iteration at $t = T$, the final objective and its gradient have to be assembled accordingly, see (5.81) and (5.83).

5.7 Remarks on the numerical implementation

As was mentioned within the previous chapter, the $\bar{\boldsymbol{F}}$-method is utilised within the structural analysis. Likewise, the adjustments regarding derivatives w.r.t. the deformation gradient within this chapter have to be considered. See Appendix A.3 regarding the $\bar{\boldsymbol{F}}$-method and its implementation in this work.

```
Global sensitivity computation

Input: $\mathbf{X}$,$\mathbf{m}_\mathrm{p}$,$\mathbf{K}_{aa}$,$\mathbf{w}$,$\mathbf{D}^\mathrm{v}$,$\mathbf{h}_n$,$\mathbf{Z}_n$

1. Loop over elements $n_\mathrm{el}$
    a) Set sensitivity switch: ssw='sens'
    b) Compute element-wise matrices $\mathbf{P}_e, \mathbf{Q}_e \longrightarrow$ Fig. 5.3
2. Assemble: $\mathbf{P}$ and $\mathbf{Q}$
3. Compute global sensitivity matrix
    $\mathbf{S}_a = -\mathbf{K}_{aa}^{-1} \cdot [\mathbf{P}_a + \mathbf{Q}_a] \quad \to \mathbf{S} = \begin{bmatrix} \mathbf{S}_a \\ \mathbf{0} \end{bmatrix}$
4. Loop over elements $n_\mathrm{el}$
    a) Set sensitivity switch: ssw='objConstr'
    b) Compute: $\mathfrak{J}_e$ and $\nabla\mathfrak{J}_e \longrightarrow$Fig. 5.3
    c) Set sensitivity switch: ssw='update'
    d) Update history sensitivities $\mathbf{Z}_{e,n+1} \longrightarrow$ Fig. 5.3
5. Assemble: $\mathfrak{J}$ and $\nabla\mathfrak{J}$
6. Assemble updated history sensitivities in history field $\mathbf{Z}_{n+1}$
```

Figure 5.2: Sketch of a global sensitivity computation of an arbitrary objective $\mathfrak{J}$.

In addition, within the previous section the global history sensitivity matrix $\mathbf{H}$ and elemental history sensitivity update $\mathbf{Z}_e$ were introduced. It was described, how an assembly for the former quantity could be achieved and numerically accounted for, while the latter was only considered at element level. Their multiplication into the matrix $\mathbf{Q}_e$ and its global assembly $\mathbf{Q}$ finally defines the change in structural response $\delta\mathbf{w}$ due to a change in history variables of the previous time step $\delta\mathbf{h}_n$. In the numerical implementation however, only the matrix $\mathbf{Q}_e$ is returned from each element and assembled globally. The element-wise contributions $\mathbf{H}_e$ and $\mathbf{Z}_e$ are directly multiplied within each element at Gauss-point level and assembled into an element quantity $\mathbf{Q}_e$. This is more beneficial, as only the matrix $\mathbf{Q}$ is specifically required for the global computation of the sensitivity matrix $\mathbf{S}$, see (5.47) and (5.63). This numeric treatment reduces overall computation time and memory requirements, as fewer variables are needed on the global level.

The numerical implementation is depicted in Fig. 5.2 conceptually, as it was implemented into AOPS, without direct specification of all the small details. It mostly aims to contextualise where the derived quantities need to be inserted into the implementation and refers to the specific equations.

```
Element sensitivity computations

Input: ssw,optsw,$\mathbf{w}_e$,$\mathbf{X}_e$,$\mathbf{m}_\mathrm{p}$,$\mathbf{h}_{e,n}$,$\mathbf{Z}_{e,n}$,$\mathbf{D}^\mathrm{v}_e$

Gauss integration loop $n_\mathrm{gp}$
  1. Compute material model response $\longrightarrow$ Fig. 5.4
  2. Compute sensitivity data according to ssw:
     a) If ssw='sens'
        i. If optsw='geom': Compute $\mathbf{P}^\mathrm{gp}_e$, see (5.39) & (5.40)
        ii. If optsw='mat': Compute $\mathbf{P}^\mathrm{gp}_e$, see (5.55) & (5.56)
        iii. $\mathbf{P}_e = \mathbf{P}_e + \mathbf{P}^\mathrm{gp}_e$
        iv. $\mathbf{Q}_e = \mathbf{Q}_e + \mathbf{H}^\mathrm{gp}_e \cdot \mathbf{Z}^\mathrm{gp}_{e,n}$, see (5.44) or (5.61)
     b) If ssw='update'
        i. If optsw='geom': $\mathbf{Z}^\mathrm{gp}_{e,n+1}$ according to (5.49)
        ii. If optsw='mat': $\mathbf{Z}^\mathrm{gp}_{e,n+1}$ according to (5.64)
     c) If ssw='objConstr'
        i. If optsw='geom': See e.g. Sect. 5.6.1
        ii. If optsw='mat': See e.g. Sect. 5.6.2
```

Figure 5.3: Sketch of the sensitivity computations on the FE level.

Local constitutive model enhancements

Input: $\mathbf{F},\phi,\nabla_{\boldsymbol{X}}\phi,\mathbf{h}_n$

1. Compute spatial damage model$\longrightarrow$ Fig. 4.3
   a) Compute additional spatial derivatives $\longrightarrow$ Appendix B
2. Compute material response$\longrightarrow$ Fig. 4.2 & Fig. 4.4
3. If `optsw`=’mat’: Nonlocal driving forces w.r.t. $\mathbf{m}_\mathrm{p}$ $\longrightarrow$ Appendix B.3
4. Compute derivatives w.r.t. history field $\mathbf{h}_n$ $\longrightarrow$ Appendix B.1
5. Compute derivatives for history sensitivity update
   a) history field $\mathbf{h}$ w.r.t. $\mathbf{F}$, see Appendix B.2.1
   b) history field $\mathbf{h}$ w.r.t. ϕ, see Appendix B.2.2
   c) history field $\mathbf{h}$ w.r.t. $\mathbf{h}_n$, see Appendix B.2.3
   d) If `optsw`=’mat’: history field $\mathbf{h}$ w.r.t. $\mathbf{m}_\mathrm{p}$, see (5.65) - (5.67)

Figure 5.4: Additional sensitivity computations within the local constitutive model.

5.7.1 Geometric implementation

In contrast to the notation in this section, the application of these sensitivities for the presented results in Sect. 6.2 are furthermore enhanced. Instead of using the coordinates of the initial geometry $\mathbf{X}$ as the design variables $\mathbf{s}$ of the optimisation, the initial geometry is defined by means of CAGD. This CAGD is defined by control points $\mathbf{p}$, which govern the underlying mesh. While in theory, the nodal coordinates of the FE mesh might be used as the design variables, the resulting challenges within the implementation and optimisation would be significant. First of all, this would drastically increase the number of design variables, as the number in the discrete geometrical setting n_X is generally much higher than the number of control points n_cp of any CAGD, i.e. generally $n_\mathrm{X} \gg n_\mathrm{cp}$ holds. Secondly, it would be very difficult to ensure a smooth and valid mesh in each iteration of the SO. Many constraints would have to be introduced to ensure this smoothness, which further complicates the optimisation. Since consideration of the CAGD is very straightforward, with means of a chain rule, this approach is chosen here. For the sensitivity matrix derived in (5.47), this results in the simple chain rule

$$\mathbf{S}_\mathrm{opt} = \mathbf{S}_X \cdot \frac{\partial \mathbf{X}}{\partial \mathbf{p}} = \mathbf{S}_X \cdot \mathbf{D}^\mathrm{v}. \tag{5.88}$$

The additional partial derivatives $\mathbf{D}^\mathrm{v}$, which correlates the geometrical description the node points of the CAGD is called the design velocity matrix. In this work, Bézier surfaces are used to describe the CAGD.

Furthermore, within the implementation, the design velocity matrix is further split into element-wise contributions $\mathbf{D}_e^\mathrm{v}$ which are forwarded to each element computation. This has highly beneficial impact on the memory requirements of the code. Since the matrices for history sensitivity update $\mathbf{Z}_e$ have to be stored at each Gauss-point at each element, reduction in their dimension drastically reduces the required storage. By forwarding a local, element-wise design velocity field $\mathbf{D}_e^{\mathrm{v}}$, the mapping of the sensitivities from geometrical description $\mathbf{X}$ to CAGD description $\mathbf{p}$ can be achieved on the local scale. Therefore, the quantities stored at each Gauss-point can be reduced from dimension $n_{\mathrm{hv}} \times n_{\mathrm{X}}$ to $n_{\mathrm{hv}} \times n_{\mathrm{cp}}$, which drastically reduces the total memory requirements.

5.7.2 History dependent algorithm

In standard optimisation procedures utilising gradient-based optimisation, e.g. in elastic problems, the computation of the sensitivity data for objective and constraints is done sequentially after finishing the complete structural analysis, see Fig. 3.2 in Sect. 3.4. That is, initially the complete FE problem is computed for the specific BVP over the time $t = t_0, t_1, \ldots, T$. Afterwards, the sensitivity analysis is performed with the information from the structural analysis at $t = T$. For history dependent materials however, this no longer is sufficient, as was previously presented in this chapter. An iterative update of the sensitivity data is required after each converged global Newton step $t \in [0,T]$. This requires an adjusted implementation regarding the calculation of this sensitivity data. The enhancement regarding the algorithmic treatment is depicted in Fig. 5.5. This enhancement replaces the sequential *Structural analysis* and *Sensitivity analysis* steps in Fig. 3.2, and replaces them with an alternating update scheme at each time load step t.

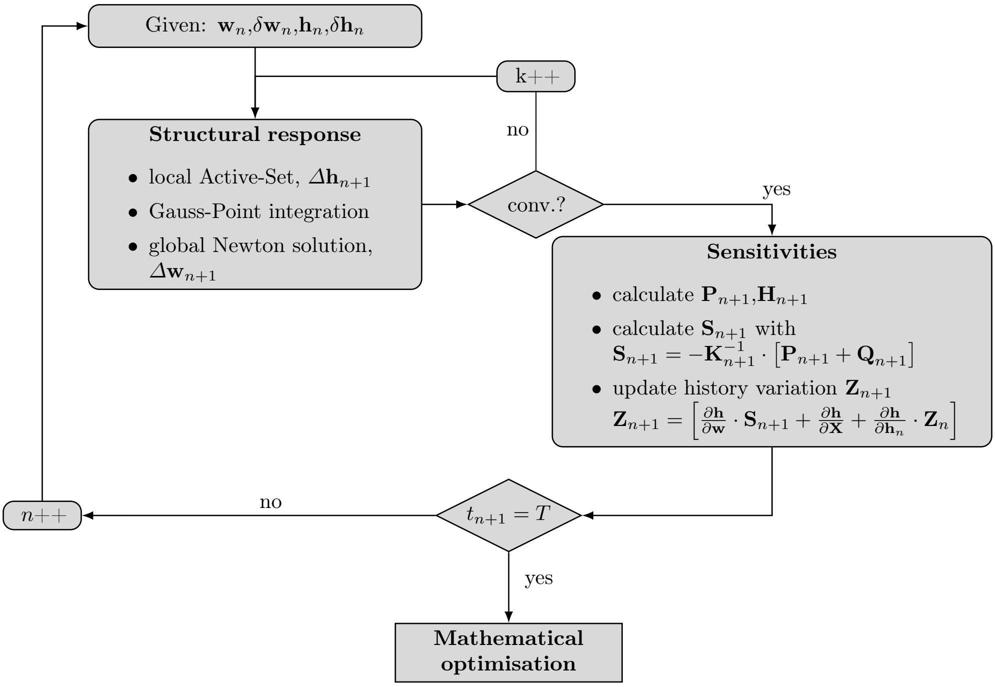

Given: $\mathbf{w}_n, \delta\mathbf{w}_n, \mathbf{h}_n, \delta\mathbf{h}_n$
k++
no
Structural response
• local Active-Set, $\Delta\mathbf{h}_{n+1}$
• Gauss-Point integration
• global Newton solution, $\Delta\mathbf{w}_{n+1}$
conv.?
yes
Sensitivities
• calculate $\mathbf{P}_{n+1}, \mathbf{H}_{n+1}$
• calculate $\mathbf{S}_{n+1}$ with $\mathbf{S}_{n+1} = -\mathbf{K}_{n+1}^{-1} \cdot \left[\mathbf{P}_{n+1} + \mathbf{Q}_{n+1}\right]$
• update history variation $\mathbf{Z}_{n+1}$ $\mathbf{Z}_{n+1} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{w}} \cdot \mathbf{S}_{n+1} + \frac{\partial \mathbf{h}}{\partial \mathbf{X}} + \frac{\partial \mathbf{h}}{\partial \mathbf{h}_n} \cdot \mathbf{Z}_n\right]$
n++
no
$t_{n+1} = T$
yes
Mathematical optimisation

Figure 5.5: Illustrative flowchart for the updates of sensitivity matrix $\mathbf{S}$ and history sensitivity update $\mathbf{Z}$ in prescribed load step n.

Given the data from previous time step $t = t_n$, the structural analysis for the BVP at t_{n+1} is iteratively solved for a new update $\Delta\mathbf{w}_{n+1}$. If the iteration converges, the sensitivities are computed. This requires initial computation of pseudo load matrix $\mathbf{P}_{n+1}$ and history sensitivity matrix $\mathbf{H}_{n+1}$. Combining the latter with the history sensitivity of the previous time step t_n, i.e. $\mathbf{Q}_{n+1} = \mathbf{H}_{n+1} \cdot \mathbf{Z}_n$, allows the computation of the Sensitivity matrix $\mathbf{S}_{n+1}$. This matrix in return is used to update the history sensitivity update $\mathbf{Z}_{n+1}$, which is stored for the next time increment $t = t_{n+2}$. If the last time step is reached, i.e. $t = T$, the mathematical optimisation can be conducted. Otherwise the counter $n = n + 1$ is increased and the procedure continues.

5.8 Numerical verification

To verify the correct derivation and implementation of the analytically derived gradients, they are compared to numerical gradients by means of the FDM. For this purpose, a cube is chosen as the test example. This rather simple structure is necessary, as computation of numerical gradients for even small problems leads to very long computation times. It is therefore not feasible to compute the numerical gradients for problems of a larger scale with the underlying code. Furthermore, due to the restrictions and assumptions made for the derivation of the analytical gradients, see Appendix B.1, an inhomogeneous load case is necessary for the test example and is depicted in Fig. 5.6. Symmetry conditions are applied requiring only an eighth of the cube to be computed, i.e. the meshed part in the figure. It is discretised with 5 elements in all three directions with total edge length of $h = w = t = 2\,\text{mm}$. The prescribed displacements reach $u_1^{\text{pre}} = 0.125\,\text{mm}$ and $u_2^{\text{pre}} = u_1^{\text{pre}}/2$ at the final load time step. The material parameters for the simulation are listed in Table 5.1. An unloading procedure is included during the middle of the simulation, to ensure correct implementation of the history update under these circumstances; this occurs during steps 9 and 21.

For the test, the CDQ is computed and compared to the analytically derived gradients. The tolerance for the perturbation value is set to $\epsilon = 1 \times 10^{-6}$. The design variables for the perturbation are the control points when testing the geometric gradients, and all material parameters when testing the constitutive sensitivities. The compared quantities are the sensitivity matrix $\mathbf{S}$, as well as the compliance $C(\mathbf{p})$ in the geometric case the, or the simulated forces $F^{\text{sim}}(\mathbf{m}_{\text{p}})$ in the material case. Even though throughout this work only the coupled case is considered during derivation of all quantities, for the implementation the elastic, plastic and damage case have to be considered. For this purpose, the respective material parameters are adjusted, such that the corresponding yield conditions are never exceeded and the chosen material behaviour case enforced. The adapted parameters are listed in Table 5.2.

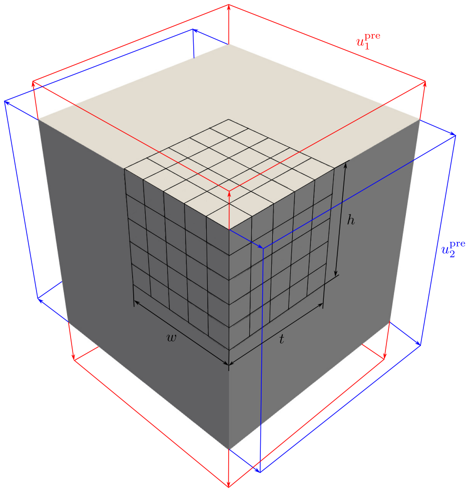


Figure 5.6: The BVP for the numerical verification of the sensitivity information.

Table 5.1: Set of material parameters for the simulations.

Symbol	Value
E/MPa	207000
ν	0.21
$\sigma_{\text{y0}}/\text{MPa}$	731.64
h/MPa	14432.08
n_{p}	0.9983
η	2.3509
ξ_{vol}	1
ξ_{iso}	0.7806
ξ_{q}	0.6443
ξ_{m}	2.3145
η_α	0.0188
$q_{\text{min}}/\text{MPa}$	17.6545
n_{d}	2/3
c_{d}/N	37.4659
$\beta_{\text{d}}/\text{MPa}$	500

Table 5.2: Necessary adjustments to the material parameters in Table 5.1 for the four different cases of material behaviour.

material parameter	elastic case	plastic case	damage case	coupled case
$\sigma_{\text{y0}}/\text{MPa}$	1×10^6	731.64	1×10^6	731.64
$q_{\text{min}}/\text{MPa}$	1×10^6	1×10^6	17.6545	17.6545
n_{d}	0	0	2/3	2/3

To quantify the difference between the numerically and analytically derived gradients, a relative error measurement between both quantities is calculated, i.e.

$$\mathbf{e}_\bullet = \frac{\|\bullet^{\text{num.}} - \bullet^{\text{analyt.}}\|}{\|\bullet^{\text{num.}}\|}, \tag{5.89}$$

where the $\bullet$ is the choice of compared quantity, i.e. $\mathbf{S}$ or gradient of the objective $\nabla\mathfrak{J}$. The superscripts num. and analyt. represent the numerical and analytical quantities, respectively. The results for both verification procedures are depicted in Fig. 5.7. The data confirms the correct implementation of the analytically derived sensitivity information as the relative error remains relatively small, staying below a value of 1×10^{-6} and sometimes reaching values of below 1×10^{-10}.

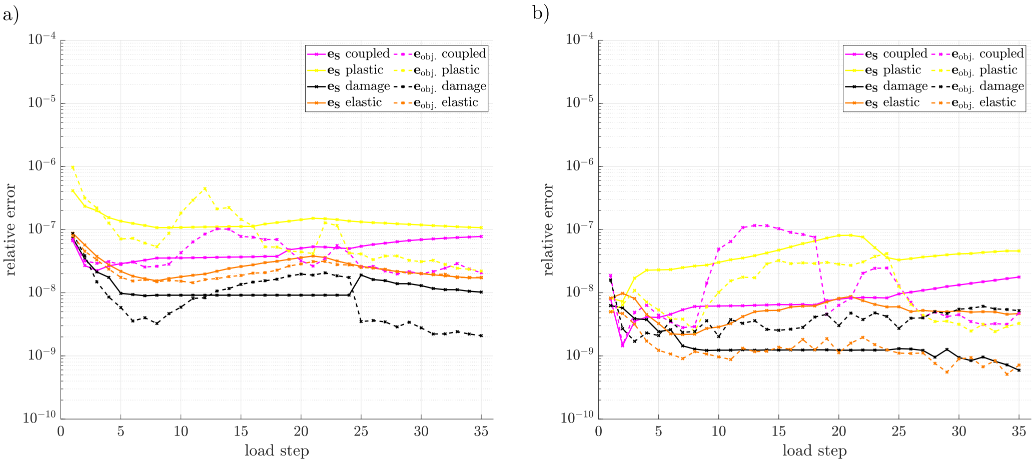


Figure 5.7: Numerical verification of sensitivity computation: a) The relative error for the geometric design variables $\mathbf{p}$ and b) for the constitutive design variables $\mathbf{m}_{\mathrm{p}}$.

Chapter 6

Damage oriented optimisation

The previous chapter provided the methodology to derive additional gradient information for the ductile damage material model and employ them in the numerical framework AOPS. The geometric sensitivities are used for shape optimisation for two different stated problems. First damage is directly optimised during shape optimisation by choosing the objective as a function directly depending on the damage variables. This objective computes the accumulated damage within the considered geometry, and the shape is optimised, such that total damage in the body is minimised. In a second problem, damage is considered only indirectly within the objective. A compliance function is optimised and structures with and without damage modelling are analysed. The material sensitivities are not used for general numerical optimisation. Instead, they are directly analysed for geometries used within parameter identification and hence conclusions drawn regarding identifiability of parameters within given geometries.

6.1 Academic damage optimisation

In this chapter, the gradient information derived in the previous chapter are utilised to optimise and analyse ductile damage problems. Due to the already existing complexity of the material model and the addition of optimisation, academic problems are computed and the numerically optimised structures analysed. The material under consideration is DP800 steel, with the material parameters successfully having been identified in [215]. The simulation model for a plate with a hole is presented in Sect. 6.2.1. Additionally, the Computer Aided Geometric Design (CAGD) model is presented, wherein the design variables and their bounds are specifically highlighted. In Sect. 6.2.2 and Sect. 6.2.3, the shape of the plate is optimised. In the former, damage optimisation is conducted, where the objective directly takes damage into account. That is, a least-square function of the nonlocal damage captures the total accumulated damage in the body and the minimisation generates shapes with reduced damage accumulation. The latter compares structures from optimisations with and without considering damage during the simulations, and hence optimisation. For this, the compliance is minimised, as it is only implicitly dependent on the damage behaviour. That is, the function can be described independently on the chosen material parameters and therefore elastic, elasto-plastic and ductile damage behaviour are modelled and optimised separately. Comparison of the results allows for a generalised statement regarding utilisation of damage modelling within optimisation problems.

The gradient information w.r.t. the constitutive material parameters are not used for optimisation. In [215], parameter identification (PI) using gradient-free methods was already conducted for the implemented material model and it was fitted to behaviour of DP800. Instead, here the sensitivity data is directly analysed. The models from the PI are utilised to compute the gradient information, which is directly analysed. This allows postulating statements regarding identifiability of certain parameters for different geometries. Since this methodology is only a prototype, this will only be briefly discussed.

Parts of the presented research are already published. In [100, 103] shape optimisation is applied to a different nonlocal damage models, while in [96, 99] shape optimisation and parameter sensitivity studies were applied to the material model considered in this thesis.

6.2 Shape optimisation

In this section two optimisation problems are discussed which are applied to the academic problem of a plate with a hole. Initially, the simulation and optimisation setup are described and a reference solution presented in Sect. 6.2.1. The following sections detail the results for damage based optimisation in Sect. 6.2.2 and compliance optimisation in Sect. 6.2.3. The optimisation problems are solved with the gradient-based method of Sequential Quadratic Programming (SQP) of the optimisation toolbox in Matlab, specifically `fmincon`, with standard parameters, such as the break threshold which finishes the optimisation.

6.2.1 Simulation model

The academic problem chosen for the following optimisations is that of a plate with a hole, due to its inhomogeneous deformation and resulting stress state. The material parameters for the simulation are taken from the source of the model, i.e. [215], and have been listed in Table 5.1. In order to utilise these parameters accurately, the simulation model is required to have comparable dimensions to the specimen used in the PI of the cited work. The simulation problem itself is depicted in Fig. 6.1, wherein, by exploiting the symmetry of the problem and applying according boundary conditions, only the meshed area is used for the simulations at hand. The plate has a thickness of $t = 1\,\mathrm{mm}$ and height, as well as width, $h = w = 10\,\mathrm{mm}$. The inner radius of the hole is equal to $r = 2.5\,\mathrm{mm}$. At the top and bottom of the plate, a prescribed displacement of $u^{\mathrm{pre}} = 0.5\,\mathrm{mm}$ is applied. The mesh itself consists of 1350 8-noded hexahedron elements, which is a sufficiently fine mesh to regularise the damage behaviour, cf. [103, 213], while simultaneously reducing overall computation time by using as small of a mesh as possible.

The mesh itself is generated by CAGD, in this case Bézier-surfaces, cf. [74], and consists of two separate patches, which conjoin at a 45° angle in the reference plate, indicated by the green line in Fig. 6.1. Certain control points of these patches are chosen as the subsequent design variables $\mathbf{s} = \mathbf{p}$ for the following optimisations and are depicted in Fig. 6.2. The dashed lines therein indicate the possible design space for the respective

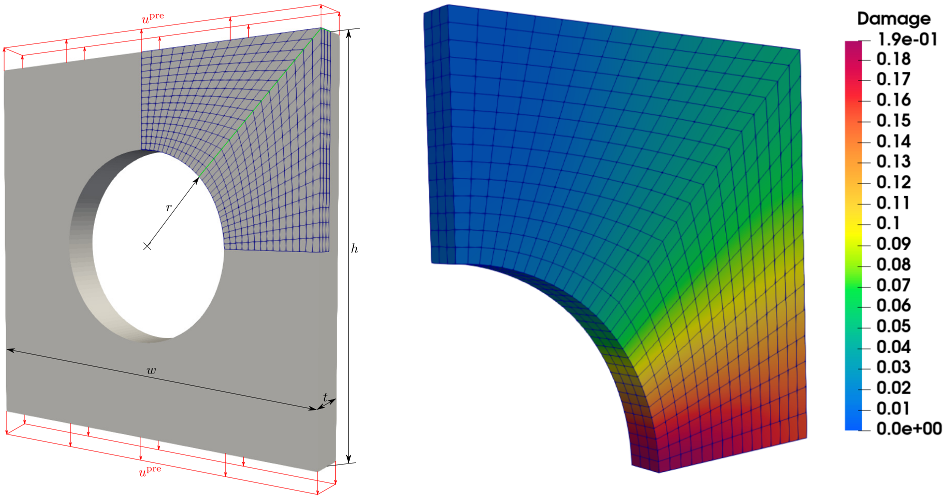


Figure 6.1: The problem definition and the reference damage distribution.

control point within the optimisation and are therefore included as the upper $\mathbf{p}_\mathrm{u}$ and lower $\mathbf{p}_\mathrm{l}$ bounds, which make up the design space of the optimisation. The geometric design is based on [98, 103], but includes also the thickness direction as a possible variation in design by allowing certain control points to vary in thickness direction as well. All design variables not lying on the symmetry surface in thickness direction are allowed to vary their coordinate in thickness direction by $\varDelta\mathbf{p} = 0.25\,\mathrm{mm}$ from their starting point. The only exception are the three control points at the right edge, which can double their thickness by $\varDelta\mathbf{p}_\mathrm{u} = 0.5\,\mathrm{mm}$, but have the same constraint when reducing the thickness. Additionally, while not further specified here, certain equality constraints apply to specific control points, which are required for the Bézier-surfaces to yield valid meshes and uphold the symmetry conditions exploited before.

The resulting damage distribution throughout the plate is displayed on the right part of Fig. 6.1 and is used as a point of reference for the following optimisation problems in Sect. 6.2.2 and Sect. 6.2.3, respectively. The maximum damage value for the reference state reaches up to $\phi^{\mathrm{ref}}_{\mathrm{max}} = 0.1893$ in the lower left corner of the bottom part of the plate. Additionally, this plate also shows small indications of damage localisation in the lower left corner of the plate, which will be elaborated on in the following sections.

6.2.2 Damage minimisation

The objective of the first optimisation problem is to reduce the overall damage accumulation within the body. The total accumulated damage is captured in the norm of the damage vector $\boldsymbol{\phi}(\mathbf{p})$, see Sect. 5.6. For sake of comparison regarding the different designs achieved, an additional constraint is added, taking the volume of the body into account. This constrains the optimiser to utilise the same amount of material in each generated geometry.

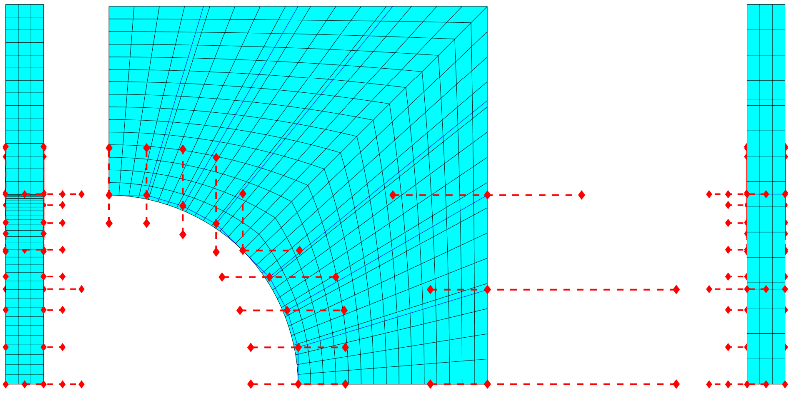


Figure 6.2: The initial CAGD of the plate with a hole and the design variables as the control points in red. The dashed lines are potential positions for the optimisation with the edges their respective bounds.

The optimisation problem therefore reads

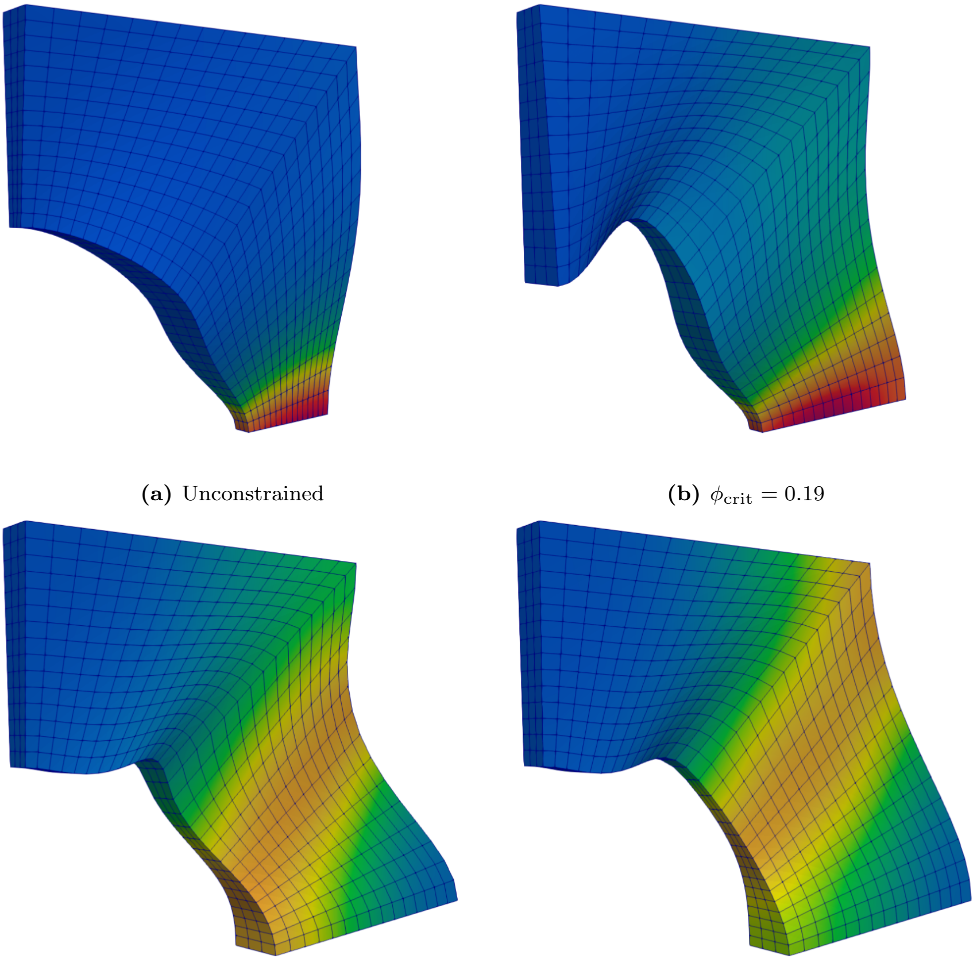
$$
\begin{aligned}
&\underset{\mathbf{p}_\mathrm{l}\leq\mathbf{p}\leq\mathbf{p}_\mathrm{u}}{\text{minimise}} \quad && \mathfrak{J}^\mathrm{d} = \|\boldsymbol{\phi}(\mathbf{p})\|^2 \\
&\text{subject to} && V(\mathbf{p}) = V_0 .
\end{aligned}
\tag{6.1}
$$

The geometry generated by the stated problem is depicted in Fig. 6.3a). The optimiser reduces the cross-section area on the bottom of the plate to its minimal thickness and width, while compensating the resulting reduction in volume by increasing the thickness around the area where both CAGD patches are connected. While this solution does minimise the objective function to a desirable degree, see Fig. 6.4 which depicts the objective over iterations, it does not necessarily generate the aimed for result. This, from a mathematical viewpoint, optimal shape leads to an early, geometrically induced, localisation of damage, compared to the reference design. This would, upon further loading with appropriate fracture modelling, lead to earlier failure of the specimen, compared to the reference design. Since the area where the damage localises is rather small, this leads to a lower objective value overall and hence the resulting design from a mathematical standpoint. However, this is not the intended (or desired) outcome of the stated, damage based optimisation problem. More desirable would be a design which not only reduces the overall accumulated damage throughout the body, but furthermore restricts damage spikes and thus may delay the localisation of damage upon further loading, which subsequently could delay failure for such a specimen.

To force the optimiser to generate such a (more desirable) design, optimisation problem (6.1) is adapted by additional nodal damage constraints, i.e.

$$
\begin{aligned}
&\underset{\mathbf{p}_\mathrm{l}\leq\mathbf{p}\leq\mathbf{p}_\mathrm{u}}{\text{minimise}} \quad && \mathfrak{J}^\mathrm{d} = \|\boldsymbol{\phi}(\mathbf{p})\|^2 \\
&\text{subject to} && V(\mathbf{p}) = V_0 \\
& && \boldsymbol{\phi}_i \leq \phi_\mathrm{crit} .
\end{aligned}
\tag{6.2}
$$

With these additional constraints, it is possible to restrict the resulting localisation arising in the unconstrained problem of (6.1), by restricting the maximum allowed global damage value $\boldsymbol{\phi}_i$ each finite element node may accumulate. The results for this optimisation problem, with varying values of ϕ_{crit}, are depicted in Fig. 6.3b)-d). The critical values ϕ_{crit} were lowered from a starting point of $\phi_{\text{crit}} = 0.19$ down to $\phi_{\text{crit}} = 0.12$, with three exemplary designs presented here. The lowest achievable solution is that with a constraint value of $\phi_{\text{crit}} = 0.12$, after which the optimiser is not able to generate structures while simultaneously fulfilling the constraints, mainly due to the chosen limits in design freedom $\mathbf{p}_\text{l} \leq \mathbf{p} \leq \mathbf{p}_\text{u}$, again see Fig. 6.2.



(a) Unconstrained

(b) $\phi_{\text{crit}} = 0.19$

(c) $\phi_{\text{crit}} = 0.15$

(d) $\phi_{\text{crit}} = 0.12$

Figure 6.3: Different optimal shapes resulting from unconstrained (a) and constrained (b)-(d) damage minimisations.

Starting with a constraint value of $\phi_{\text{crit}} = 0.19$, the resulting geometry already differs strongly from the unconstrained problem (6.1). The lower cross-section gets reshaped by increasing the overall width, but the overall thickness in this area still gets reduced. To compensate in volume, the central hole gets reshaped and certain parts of the plate are varied in thickness. The right side of the plate starts to curve outwards, which is opposite to the unconstrained problem, where the curvature turns inwards. Of note here, is that this bounded critical value is higher than the initial maximum value $\phi^{\text{ref}}_{\text{max}} = 0.1893$. Still, these additional constraints nudge the optimiser into a search direction, which leads away from the (undesirable) solution of the unconstrained problem and into the new (more desirable) design. Even with these constraints, geometrically induced localisation of damage, while not as pronounced as in the unconstrained problem, is still detectable on the lower part of the plate. Reducing the critical damage value further to $\phi_{\text{crit}} = 0.15$ leads to a critical switch in design of the lower part of the plate. The cross-section at the bottom gets even larger, by further increasing its width which is amplified by additionally increasing the thickness as well. To compensate in volume, and therefore fulfilling the equality constraint, the central part of the plate gets thinner. The combination of both these design changes leads to a shear-like damage zone in a 45° angle from the bottom left to the middle right side of the newly generated plate. All previous designs, including the reference problem, had this damage zone lying horizontal at the bottom edge of the plate, leading to damage localising behaviour. Additionally, the right side of the plate has an even more pronounced curvature, which started to develop in the previous constrained problem. Of note for this problem is, even though the damage is bounded to $\phi_{\text{crit}} = 0.15$, the maximum damage only reaches a value $\phi^{\text{max}}_{0.15} = 0.1241$. To further reduce this maximum damage, the upper bound is further reduced to $\phi_{\text{crit}} = 0.12$ in the last design presented in Fig. 6.3d). To fulfil this new constraint, the optimiser pushes the thinnest area of the plate a bit upward, just below the area where the two Bézier surfaces are connected. This in turn moves the aforementioned damage zone upwards as well, which ends just below the top-right corner of the plate. Additionally, the right side no longer curves inward and, as such, the leftmost point on the right surface of the plate is at the top right corner of the plate. This finally leads to a maximised area of the damage zone and with this, it is possible to bound all damage values to the prescribed critical value of $\phi_{\text{crit}} = 0.12$. Due to the broad area where damage accumulates in those latter two examples, the geometrically induced localisation of damage can be prevented, which would theoretically allow these specimens to be subjected to larger deformations, before failure occurs.

With the addition of the damage constraints, as would be expected, the respective objective values obtainable at the end of each optimisation get significantly increased, see Fig. 6.4. The unconstrained problem is able to reduce its objective value by 66.7% from $\mathfrak{J}^{\text{d,init}}_{\text{unconstr}} = 14.3414$ to $\mathfrak{J}^{\text{d,opt}}_{\text{unconstr}} = 4.7746$. This is possible, as already mentioned, due to geometrically localising the damage accumulation in a relatively small area. With the inclusion of the additional constraints in damage, the objective value cannot be reduced as significantly. For the analysed problems in this work, the lowest achievable objective reached is $\mathfrak{J}^{\text{d,opt}}_{0.19} = 7.6292$ for the highest damage constraint, and subsequently $\mathfrak{J}^{\text{d,opt}}_{0.12} = 9.1196$ for the lowest critical damage bound. Nonetheless, this is still the equivalent of a reduction between $46.8\% - 36.4\%$ of accumulated damage in the considered problem. Furthermore, while the unconstrained problem is able to find an optimum after only 4 iterations, the additional $n^{\phi}_{\text{np}} = 1350$ damage constraints, while not necessarily active at all time, in the other problems lead to many further iterations for the optimisation to converge. In the case of the lowest possible critical damage value of $\phi_{\text{crit}} = 0.12$, up to 45 iterations are necessary to finally converge. Of note is the iterative behaviour of the magenta curve in Fig. 6.4. While the $\phi_{\text{crit}} = 0.12$ and $\phi_{\text{crit}} = 0.15$ problem end up very similar in final geometric design, the iterative process for the former is more difficult, as the optimiser has to take a different path, wherein the more difficult constraint has to be fulfilled throughout the optimisation. As such, the optimiser steers towards a local minimum up to iteration 25 which diverges strongly from the final optimum. At that point a huge shift in shape occurs which steers the optimiser to the final design presented in Fig. 6.3d).

Additionally depicted in Fig. 6.4 are the resulting load-displacement curves for each optimisation problem. The red curve depicts the initial, reference design. While all other designs yield softer responses, even though the accumulated damage throughout the body is reduced, they still show some interesting characteristics. First of all, the unconstrained problem shows a significantly softer response. After a load of $F = 2\,000\,\text{N}$ the geometry starts losing its load-bearing capabilities and a downward trend in its curve is visible; a result of the geometrically induced damage localisation. This happens after a displacement of $u = 0.125\,\text{mm}$, while in the reference design this happens only after $u = 0.33\,\text{mm}$ at a maximum force of $F = 4\,660\,\text{N}$. The constrained problems show a generally more beneficial behaviour. With a high critical damage bound (orange curve), the design behaves more like the unconstrained one, due to also showing some localisation behaviour, but reaches a higher maximum force and at a higher respective displacement. Of interest are the additional two curves, with $\phi_{\text{crit}} = 0.15$ and $\phi_{\text{crit}} = 0.12$, respectively. The designs generated by these optimisations showed a broadened, diagonal damage distribution in the body. This distribution leads to a slower loss in load-bearing capacity by delaying the localisation of damage. Moreover, the generated shapes do not show any downward slope of the load-displacement curve within the analysed area. It is to be expected that, upon further loading, the latter design surpasses the reference design in its load-bearing capacity due to delaying the critical damage localisation. However, to generally generate stiffer components, a compliance-like optimisation would be beneficial, which is focussed on in Sect. 6.2.3.

Finally, the previously described localisation behaviour of the respective shapes can also be identified when plotting the accumulated damage, i.e. objective $\|\boldsymbol{\phi}(\mathbf{p})\|^2$, over the applied displacement u, see Fig. 6.5. The unconstrained shape shows an initially steep increase in damage accumulation up to approximately $\|\boldsymbol{\phi}(\mathbf{p})\|^2_{\text{unconstr.}} = 3.15$ at $u = 0.2\,\text{mm}$, after which the accumulation lessens significantly. The same general behaviour is also present in the constrained problem with $\phi_{\text{crit}} = 0.19$, while the significant change in the slope does not occur. The remaining two problems show a different behaviour in damage accumulation. The damage evolution starts off with a gentle incline of the curve, compared to the unconstrained behaviour. At approximately $u = 0.15\,\text{mm}$ the slope of the curve increases slightly and afterwards the accumulation of damage behaves almost linearly, showing no immediate sign of localisation.

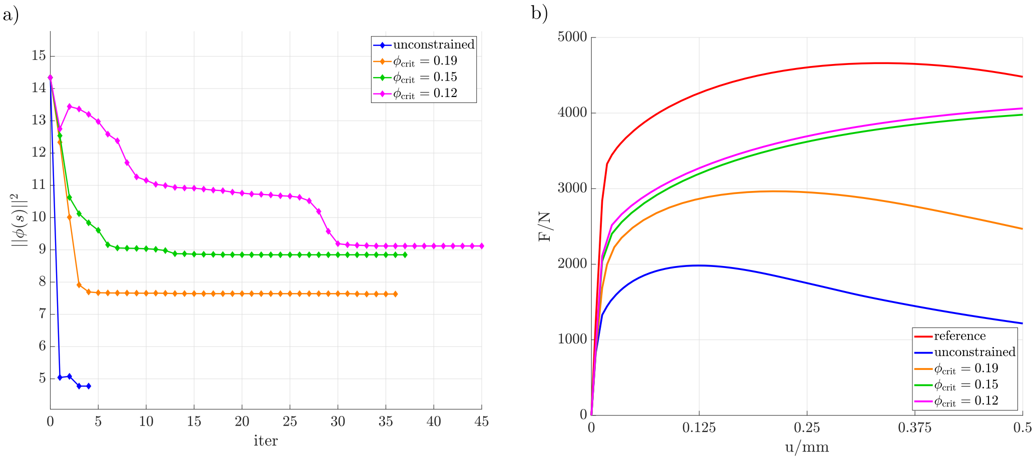


Figure 6.4: a) Objectives for different damage optimisation problems over iterations and b) load-displacement curves.

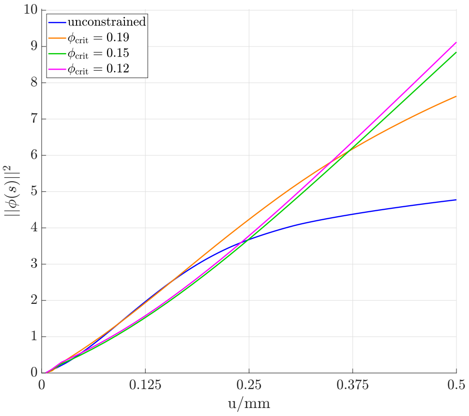


Figure 6.5: Damage accumulation over displacement for each optimised shape.

6.2.3 Compliance optimisation

Following a direct damage optimisation, one could now instead use the damage states simply as a constraint to limit the damage evolution when conducting a compliance optimisation, similarly done in [103]. However, this shall not be the aim of the presented research. Instead, in this second optimisation problem, the potential benefit of the utilised damage modelling is analysed and reviewed, whether the additional computational complexity shows sufficient gain in generated results. To allow comparison, a single optimisation problem is subjected to three different cases for the underlying material model derived in Chap. 4:

1. A purely elastic case, where the thresholds for damage and plastic initiation are set arbitrarily high to prevent their respective evolution and enforce elastic behaviour.

2. An elasto-plastic case, wherein only the damage threshold is manipulated to prohibit damage evolution.
3. An unaltered case, which leads to the coupled elasto-plastic damage behaviour.

While theoretically a fourth case, i.e. a case where only the plasticity is disabled and uncoupled damage evolves, is possible, this case is not presented here. Due to the set of chosen material parameters, the damage variable gets too high in this fourth case, which in turn leads to the optimiser failing to find a better solution than the initial guess and therefore results in no change in design. The necessary adjustments to the material parameters of Table 5.1 are summarised in Table 6.1 for the three analysed cases.

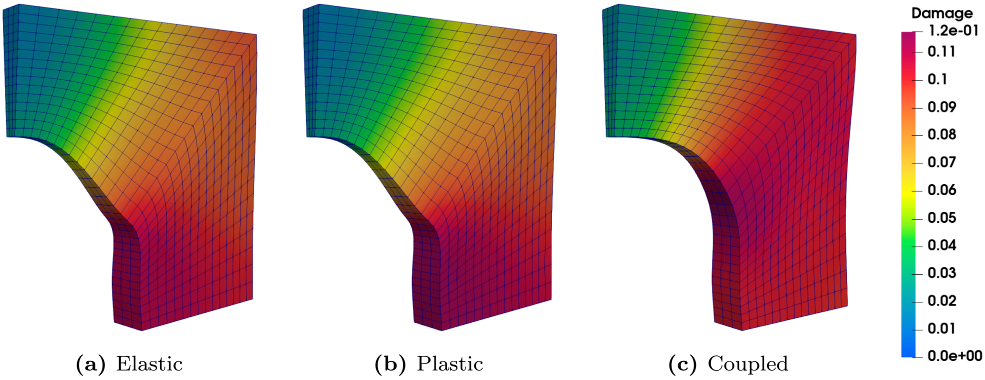
Table 6.1: Necessary adjustments to the material parameters in Table 5.1 for the three considered cases.

material parameter	elastic case	plastic case	coupled case
σ_{y0}/MPa	1×10^6	731.64	731.64
q_{min}/MPa	1×10^6	1×10^6	17.6545
n_{d}	0	0	2/3

To compare the three cases mentioned above, an objective function is necessary which not directly depends on the underlying material model in its definition. The previous damage based objective function in Sect. 6.2.2 is therefore not suitable, as no damage values are available in the elastic and plastic problem. Subsequently, the compliance $C(\mathbf{p})$, as described in Sect. 5.6, is chosen for this comparison instead, leading to the generalised optimisation problem

$$\begin{aligned} &\underset{\mathbf{p}_{\mathrm{l}} \leq \mathbf{p} \leq \mathbf{p}_{\mathrm{u}}}{\text{minimise}} && \mathfrak{J}^{\mathrm{C}} = C(\mathbf{p}) \\ &\text{subject to} && V(\mathbf{p}) = V_0 . \end{aligned} \tag{6.3}$$

Due to the choice of objective function, the material model does not directly influence the stated optimisation problem. However, the underlying material behaviour influences the resulting values of the compliance and therefore newly generated geometries. After optimisation, the resulting optimal geometries of the purely elastic and the elasto-plastic problems are subjected to simulations with the unaltered, coupled material behaviour. This in turn allows direct comparison regarding their actual stiffness, when accounting for damage accumulation, as well as their resulting damage distribution. The resulting optimised geometries and their respective damage distribution are presented in Fig. 6.6. The results from the purely elastic and the elasto-plastic behaviour are almost identical and not quite distinguishable to the naked eye. Both geometries reduce the thickness and overall width of the top-left area in order to shift that material to the lower part of the plate, resulting in a stiffer behaviour. The hole itself changes into an oval shape with the lower part being the thickest part of the newly generated structure. When directly comparing the elastic and the plastic resulting design, while not directly visible in the figure, the elastic shape is a bit wider on the lower part of the plate, which has to be compensated by reducing the thickness in the area where both CAGD patches meet. This



(a) Elastic **(b)** Plastic **(c)** Coupled

Figure 6.6: Damage distribution for the respective shapes of the compliance optimisation.

leads to a slightly higher peak damage value for the plastic shape in the lower left corner of the cross-section area. The coupled case however yields a completely different, new design. Instead of widening the bottom cross-section to maximise the overall cross-section area in loading direction, the material is shifted to overall thicken the new shape in a general sense. The oval shape of the central hole is more pronounced in this design, as the top-left area of the plate is reduced to the minimal allowed width as well as thickness, as defined by the boundaries of the respective control points. Overall, this leads the plate with the coupled material model to generate a structure, which better homogenises the damage distribution throughout the body. This prevents the localisation of damage that arise in the lower left corner in the other designs, which is similar, but not as pronounced, as in the reference design. Additionally, this design behaves similar to the results of the damage constrained problem in the previous section. That is, a diagonal like damage zone is visible in a 45° angle going from centre left to top right. It is not as pronounced as in the previous section for the damage constrained problems, but still detectable. This is not happening in the other two designs, which show a damage distribution that is more akin to the reference plate, i.e. high values in the lower left corner and reduction in damage outward from that point on.

Table 6.2: Initial and optimal compliance values for their respective cases.

	elastic	plastic	coupled
C^{init}	-7081.551	-73.4209	-16.2577
C^{opt}	-8563.084	-86.6223	-23.7402
$C^{\text{init}}_{\text{cpld}}$		-16.2577	
$C^{\text{opt}}_{\text{cpld}}$	-23.2432	-22.9930	-23.7402

The resulting numerical values of the compliance objective are listed in Table 6.2. The first two rows list the initial and optimal compliance values for their respective, unaltered cases. As expected, the elastic values are overall the highest, with the plasticity already reaching lower values by a factor of 100, since the hardening modulus h is significantly smaller than the elastic modulus E. With the addition of damage in the coupled case, this value is again reduced by a factor of 4, since the accumulation of damage additionally lowers the overall stiffness of the body and therein the capacity to capture the mechanical energy. More interesting, for the sake of comparison of these three cases however, is the coupled compliance values C_{cpld}. For these values, the coupled material model is applied to the geometries of the elastic and plastic case and the resulting compliance values calculated. Since the initial geometry is the same in all cases, the result is equal as well. Unexpectedly, the optimal values are quite similar, with the variance to the coupled result being only approximately 2.1% for the elastic result and 3.25% for the plastic case, respectively. First, this again highlights the similarity of the elastic and plastic design, even though the small, almost not distinguishable differences have a higher impact onto the objective than would be expected. Second, these values indicate, that while the coupled case generates a more pronounced design regarding the damage distribution, the resulting compliance is not significantly higher than by simply optimising the problem for a purely elastic case.

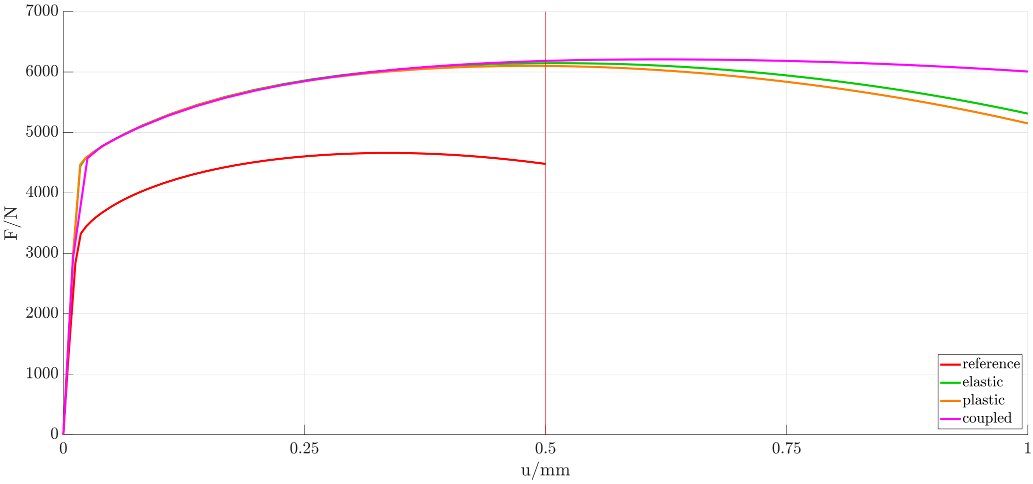


Figure 6.7: Load-displacement curves for the compliance optimisation. The vertical red line indicates the maximum applied displacement during the optimisation.

This behaviour is also identifiable when comparing the resulting load-displacement curves, see Fig. 6.7. As expected, all generated designs outperform the initial reference problem by a huge margin; able to withstand 1.5 kN more force at the same applied displacement. However, at the point of optimisation, i.e. $u^{\text{pre}} = 0.5\,\text{mm}$ indicated by the vertical red line, the differences in the three optimal designs is nearly not distinguishable. Nonetheless, comparing the three coupled compliance values in the last row of Table 6.2, the coupled optimisation results in the highest reaction force at $F^{\text{react}}_{\text{cpld}} = 6\,189\,\text{N}$, the elastic design results in $F^{\text{react}}_{\text{el}} = 6\,150\,\text{N}$ and the plastic problem results in the lowest reaction force at $F^{\text{react}}_{\text{pl}} = 6\,104\,\text{N}$.

However, when looking closely at the slopes of the respective curves, one can see that while these three curves reach nearly identical values at the point of optimisation, their underlying behaviour is not necessarily identical. That is, the elastic and plastic curves reached, or almost reached, their highest point, indicating that localisation of damage may start to occur. That would indicate a loss in load-bearing capacity and therefore a drop in the load-displacement curve is to be expected. This behaviour is not visible in the curve of the coupled case and its slope is still positive at $u^{\text{pre}} = 0.5\,\text{mm}$. This prompted to further load the geometries to a new displacement of $u^{\text{pre},+} = 1\,\text{mm}$ to analyse, how these structures behave under extended loading.

As expected from the mentioned observation, the slope of the curves from the elastic and plastic problems start to trend downward, while the coupled design is better at handling further loading. For the latter design, the slope becomes negative at around $u = 0.63\,\text{mm}$ and shows a far less severe downward trend. This highlights the benefit of running optimisations with a coupled damage material model. While the generated structure does not vastly outperform the simpler problems, the underlying consideration of damage results in a more damage tolerant structure, as it accounts for the damage behaviour, and therefore is able to withstand more loading and therefore damage accumulation. This knowledge is beneficial when taking into account safety factors that are often applied to the designing process of parts, in order to handle unforeseen extra loads. Utilising accurate damage models while conducting optimisation, may allow reduction in these safety factors, since possible, unforeseen additional loads are better absorbed for these kinds of generated geometries.

6.3 A prototype response sensitivity study

In this section, the derived sensitivities for the constitutive material parameters are utilised. However, instead of performing PI, the gradient information are analysed directly. In the original source of the material model, cf. [215], a thorough PI using digital image correlation has been conducted. Hence, further identification is not required. The method presented here is a prototype idea to give additional insight into the problem. With further research, this might lead to enhancements in effective PI and yield further insight regarding identifiability of material parameters for considered the boundary value problems (BVP).

For further analysis of the problems solved in [215], sensitivity analysis was applied to the BVPs of the simulated problems, i.e. the notched plate and the plate with a hole, see Fig. 6.8. Herein, the same identical BVPs are computed, wherein the data has been provided by the original authors. Within the PI in the source, the applied load curves within the BVP included loading and unloading steps. This is required to accurately identify the parameters which influence damage, since damage evolution has a direct impact on the elastic material behaviour.

The sensitivities are computed according to (5.84)-(5.86), i.e. the simulated reaction forces F^{sim} are of main concern. Note, that objective $\mathfrak{J}^{\text{F}}$ in (5.81) is not the considered function in this section, as such (5.87) is not required in this implementation. The sensitivities of the simulated reaction forces F_t^{sim} are computed for each material parameter $\mathbf{m}_{\text{p}}$ at each pseudo time step t, i.e. each load step. The results are depicted in Fig. 6.9 and Fig. 6.10 for the notched plate and plate with a hole, respectively. Each column depicts the sensitivity of the respective parameters has onto the simulated forces F^{sim} of the respective load step. Each row correlates with a time load step t, starting at $t = 1$ at the top and $t = T$ at the bottom. Since the absolute values of the sensitivity data very strongly differ in dimension, the presented data is normalised, i.e. each column is divided

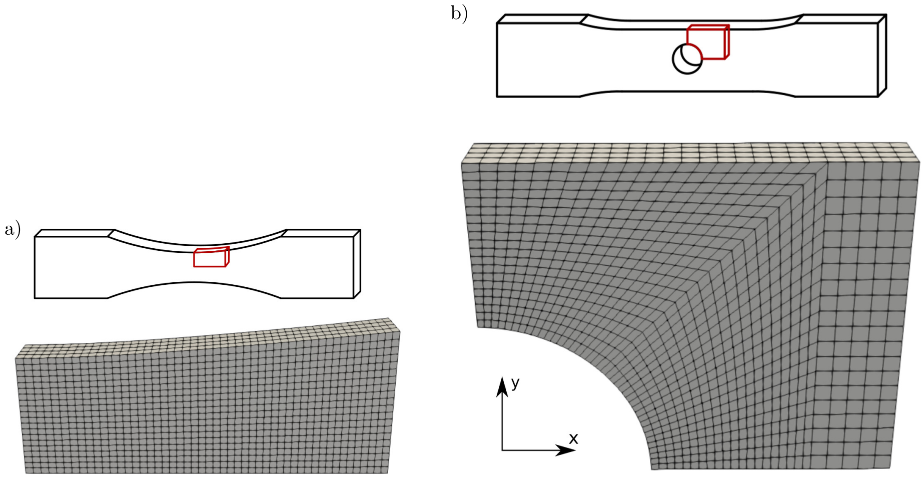


Figure 6.8: Simulation models from [215] used in the analysis of the response sensitivity data. a) The notched plat and b) the plate with a hole.

by the absolute maximum value in each column such that

$$\delta_{m_i}\mathbf{F}^{\mathrm{sim,norm}} = \frac{\delta_{m_i}\mathbf{F}^{\mathrm{sim}}}{\max(\left|\delta_{m_i}\mathbf{F}^{\mathrm{sim}}\right|)}. \tag{6.4}$$

The raw data can be found in Appendix C.2. While this normalisation allows the analysis of the respective parameters onto the simulation, this approach no longer enables direct comparison between the material parameters, since the qualitative values are no longer part of the visualised data.

Within the considered and depicted sensitivities, the respective figures only take the non-elastic parameters into consideration. Since the elastic parameters are rather straightforward to identify, they are not a concern in this study. Furthermore, β_d is omitted, as it only serves as a penalty parameter which couples local and nonlocal damage. The parameters ξ_vol and n_d are also omitted, as they are held constant in the original PI, cf. [215]. The figures additionally depict loading and unloading steps. Horizontal black lines frame a dashed black line. These are the unloading steps, wherein the first black line is the start of unloading, the dashed line the end of unloading, i.e. the turning point, and the second black line is the point where loading starts to surpass the previously applied max load again. The detailed load paths for the BVP can also be found in Appendix C.2. Of note is the extended number of load steps for the notched plate. Due to convergence issues, the time step size within the simulations is reduced drastically, leading to many iterations with little displacement increments.

Even with these restrictions, and the preliminary nature of this prototype approach, certain conclusions can be drawn from these results. First of all, some parameters show drastic sensitivity alterations between the two analysed BVPs. The yield stress $\sigma_{\mathrm{y}0}$ for example shows initially very identical, and expected, behaviour. That is, before plastic yielding the sensitivities are zero and subsequently are at their maximal at the point of

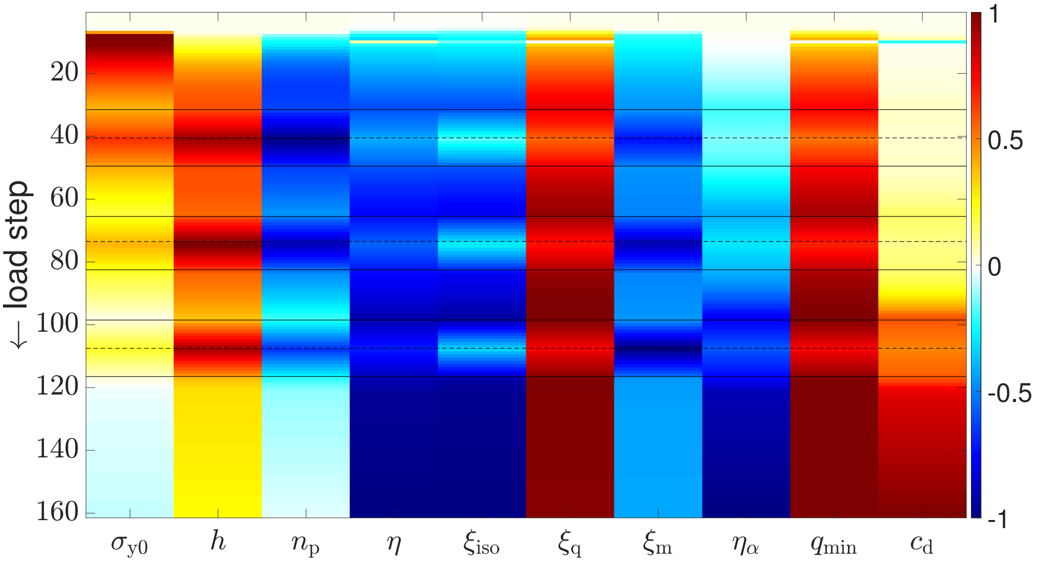


Figure 6.9: Normalised material sensitivity data for the notched plate. Load steps increasing from top to bottom.

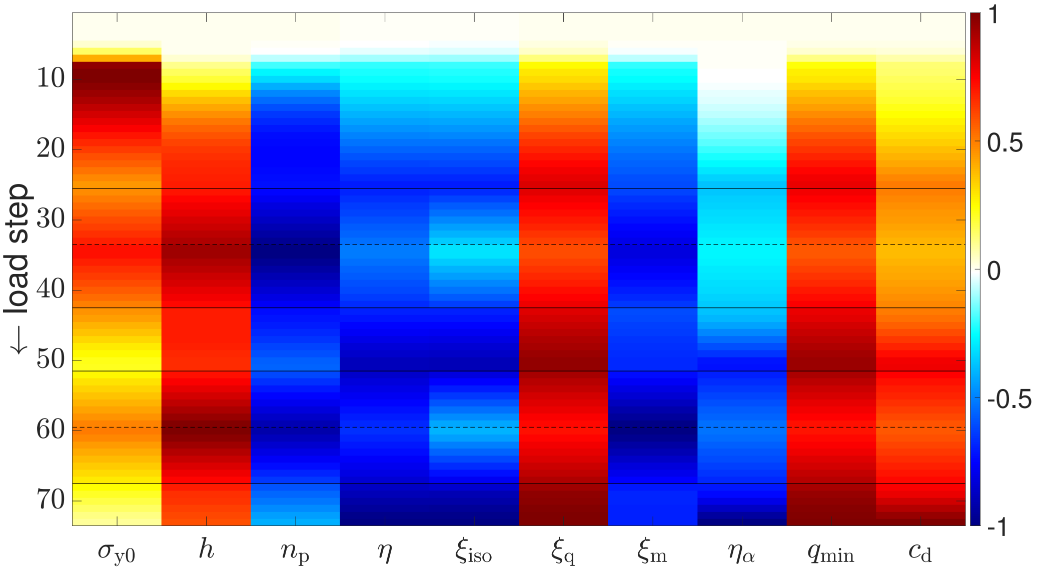


Figure 6.10: Normalised material sensitivity data for the plate with a hole. Load steps increasing from top to bottom.

initial plastic yielding. Afterwards, their influence on the simulated reaction forces F^{sim} is getting smaller. Due to plastic yielding and the evolution of damage, the sensitivity of this material parameter decreases with increased load, which is expected too. Herein, the two BVPs show a different behaviour. The influence of this parameter within the notched plate gets significantly smaller, i.e. the value is close to zero, while the influence within the plate with a hole is still not quite insignificantly small at a factor of 0.25. This same trend is identifiable for the other plastic parameters h and n_{p}.

The opposite behaviour is observed for the damage threshold $q_{\min}$, which governs the onset of damage. This quantity increases with increasing load. Similarly, this is also expected, since with growing damage this quantity is more important to further govern the damage onset due to the coupling to the current damage state via the function $[1 - f^q(d)]^{n_{\mathrm{d}}}$. Generally, an inverse behaviour between $\sigma_{\mathrm{y}0}$ and $q_{\min}$ is observed for the plastic and damage threshold in both geometries.

As an early conclusion from these results, it could be more beneficial to utilise early sections of an experiment for the notched plate to identify the plastic variables and latter sections for the damage threshold $q_{\min}$. Using this information and adapting the weight parameters w_t for the objective $\mathfrak{J}^{\mathrm{PI}}$ in (5.80) accordingly, identification of these parameters might yield better results by isolating desired material responses.

Another key parameter in the underlying model is the regularisation parameter c_{d}. Generally, this parameter is motivated by the crack width or width of the damage area within a given problem. However, its consideration within the numerics has a very pronounced effect on the mesh-independence of the material model. Therefore, its identification is generally not straightforward, as hinted at in [215]. The results from the response sensitivity study also show that both geometries depict huge differences in their results. In the notched plat this parameter shows barely any impact in the first half of the simulation and only shows drastic increase in sensitivities towards the later half. Within the plate with a hole, the sensitivities of this parameters start to increase early on and rise steadily, not showing the sudden change observed in the notched plate. This behaviour might indicate, that to better identify this regularisation parameter, only considering the latter half of the notched plate might be generally advised.

Chapter 7

Conclusion of Part I: Academic challenges

In Part I of this thesis, the academic viewpoint of damage optimisation has been considered. In Chap. 4, a nonlocal damage model and its numerical implementation have been presented. The necessary assumptions regarding the kinematics in a viewpoint of finite deformations have been introduced and the balance and thermodynamic laws summarised. The nonlocal description and implementation of the material model enables a gradient-enhanced regularisation of the damage evolution, which results in mesh-independent simulations. These are critical when considering structural optimisation, as therein mesh sizes might change which could result in inaccurate optimisation results for a mesh-dependent model. Finally, specific handling of the numerical implementation by means of the finite element method (FEM), as well as an active-set strategy for the couple behaviour of the constitutive model have been depicted.

To enable structural optimisation with gradient-based methods, in Chap. 5, the material model has been enhanced by conducting an analytical sensitivity analysis. A variational concept has been applied, which enabled an efficient and elegant method to derive the continuous gradient information without prior discretisation. For the geometrical description, an enhanced kinematic viewpoint has been utilised, as it allows decoupling the implicit dependencies of geometry and deformation to ease with derivation of the gradients. The sensitivity analysis has been applied for geometry and constitutive material parameters as possible design variables. Since the underlying material model utilises history dependent variables to store previous dissipative load steps, their contribution within the sensitivity analysis and numerical implementation had to be considered as well. This consideration has been emphasised, together with other exemplary implementation details, in the numerical treatment of the derived gradient information. The discrete quantities have been incorporated into Academic Optimisation Problem Solver (AOPS) to enable optimisation of damage. Afterwards, the implemented analytical gradients have been compared to numerically computed ones, which were gathered by means of the finite difference method. Finally, the objectives and constraints, which have been used in the computed results, have been presented.

The methodology has been applied to structural optimisation of a plate with a hole for two optimisation problem in Chap. 6. A direct damage optimisation has been conducted and damage tolerant shapes have been generated. By additionally constraining each respective FE node regarding its damage value, the resulting geometry can be significantly altered and early damage localisation prohibited. Additionally, a compliance minimisation has been conducted for different material properties, i.e. elastic, elasto-plastic, and ductile damage. The results show, that considering damage within said optimisation yield better and more damage tolerant structures. This emphasises the benefit of using complex material models within an optimisation procedure. Finally, the material sensitivities have been analysed, instead of performing a parameter identification (PI). A gradient-based PI with these sensitivities can be found in Appendix C.1.1. Utilisation of this method allows insight onto identifiability of material parameters for a given shape of test specimen. Most importantly, differences in identifiability of specific parameters like the regularisation parameter c_d can be made. Certain geometries show a higher sensitivity for this parameter which might indicate that such a shape is better at identifying said parameter.

Three main conclusions can be drawn, which also apply to other types of optimisations. These highlight the general novelty of the presented research.

1. One has to be careful stating an optimisation problem, which only considers damage in the objective. In structural optimisation this might lead to geometrically induced early damage localisation, generated due to the optimisation. Since from a mathematical viewpoint, high damage in a smaller area might be less in overall quantity than moderate damage distributed evenly over a moderate area.
2. Consideration of damage within the optimisation allows generating structures, which initially behave similar to geometries generated from a simpler model description. However, the resulting geometries allow for more damage tolerance during further loading. Extrapolating these results to other theoretical applications highlights that considering damage within the underlying model description might yield significant benefits, such as additional reserves of damage tolerance during unforeseen loads. In manufacturing of critical components for example, this allows reduction of safety margins as the optimisation creates inherently more damage resilient structures.
3. Sensitivity data for a given model has its extended applications beyond its use in gradient-based optimisation. The data can be analysed directly and conclusions drawn. For the constitutive sensitivities for example, this allows analysis of geometries used for PI.

However, even with these results, the outlined method has its drawbacks. Mainly, these come up if forming processes are considered. These processes require contact mechanics to accurately model the specific processes. To overcome this shortcoming, another optimisation environment is motivated and presented in the following. Part II: Industrial challenges will therefore take a closer look at optimisation of damage within selected forming processes.

Part II:
Industrial challenges

Chapter 8

Optimisation of forming processes

This chapter gives a brief introduction and motivation for the undertaken optimisation of industrial forming processes. Compared to the problems solved with AOPS, the boundary conditions, i.e. contact mechanics, within these industrial applications require an alternative methodology to perform mathematical optimisation, i.e. IOPS. Forming processes are discussed and the necessary background regarding the specific processes, which are considered in this thesis, is outlined. Finally, possible approaches to optimise forming processes are presented. The advantages and disadvantages are highlighted and the final choice is motivated.

8.1 Collaborative research centre TRR188

In the previous Part I of this work, an approach was described which enables the efficient numerical (shape) optimisation of structures, such that the damage accumulation under the same load is reduced in the optimised structure. Utilisation of analytically derived gradients enables an efficient implementation in the numerical framework and, with gradient-based optimisation methods, a swift generation of new structures. This method was applied in the presented framework AOPS. However, this derivation of gradients by means of analytical methods is a significant challenge. This increases further when multiple material models are considered and the additional complexity of forming processes is added. The variational approach could swiftly and efficiently handle the presented academic problems. However, forming processes represent a more difficult challenge, due to the increasing numerical complexity of the simulations. To resolve this issue the additional framework IOPS is created, which enables optimisation of forming processes.

One critical factor to consider in the modelling and simulation of forming processes is the inherent requirement of contact mechanics, i.e. the interaction of two or more separate bodies within the finite element (FE) environment [235]. While contact mechanics is an established field in conventional FE environments, their efficient implementation and hence its robustness are not straightforward to handle in non-commercial code [237]. Additionally, contact mechanics introduce a strict discontinuity into the mathematical description, making the derivation of analytical gradients more challenging [132, 136, 217].

The requirement of simulating and optimising forming processes is motivated by the funding of the research presented within this thesis. The investigations were conducted as project C05 within TRR188, a collaborative research centre (CRC) funded by the DFG (Deutsche Forschungsgemeinschaft) [224]. TRR188 is structured such that different fields of research, i.e. process technology, characterisation and modelling, conduct their own principal investigations while simultaneously cooperating in their work and supporting other projects with their results. The aim is to improve the knowledge regarding damage in forming processes by combining the investigations in these three fields of research. Consequently, as the modelling project of area C which conducts numerical optimisations, project C05 had to directly take forming processes into consideration.

Due to time constraints and the desire to cooperate with other projects within TRR188, the choice was made to follow a different approach, i.e. AOPSis exchanged for IOPS, when trying to optimise forming processes. Instead of deriving the gradients of a chosen material model by means of analytical sensitivity analysis, a more hands-on approach is chosen. While a derivation of analytical gradients for the discontinuous contact problem is possible, e.g. by means of a subgradient method [182], weak derivatives [72], or using stochastic modelling [53], these methods are not applied here. A general overview regarding optimisation of nonsmooth problems can be found in [113]. The alternative choice of optimisation chosen in this thesis enables the mutual pursuit of the academic challenges presented in Part I of this work, as well as providing the possibility of directly optimising the forming process researched within TRR188.

8.2 Introduction to forming processes

This section briefly introduces the reader of this thesis to certain forming processes in general. According to DIN 8580 [64], manufacturing processes are categorised into six groups:

1. shaping (e.g. casting),
2. forming (e.g. bending, extrusion),
3. subtractive manufacturing (e.g. milling, machining),
4. additive manufacturing (e.g. welding, 3d-printing),
5. coating,
6. heat treatment.

Forming processes themselves can further be categorised into different subgroups. One such categorisation is achieved by the choice of the dimension of the manufactured part. In bulk metal forming like extrusion or swagging a thick-walled structure is reshaped. In contrast, sheet metal forming like bending, deep-drawing and rolling utilise or create thin-walled structures. Another categorisation aims to differentiate between the working temperatures. The three fields are defined as cold forging at room temperature, warm forming below the recrystallisation temperature and hot forging above the recrystallisation temperature. One specific highlight is the benefit of work hardening which occurs during cold forging. The induced plastic deformation below the recrystallisation temperature leads to an increased density of dislocations and therefore increased stiffness. Under certain circumstances this can be exploited to improve the quality of the formed part [45, 176].

In subtractive manufacturing process, like machining for example, the component is created by the controlled removal of material from an initial larger piece. Therefore, only a certain percentage of the total material is used to create a desired part. In contrast, forming processes utilise 100 % of the material during production. Consequently, in the current environment where reduction of waste in material and energy is a crucial objective, the prospects of efficient forming processes are more important than ever. In addition to the efficient utilisation of the material, certain beneficial aspects of forming processes can be exploited. Since the material within a part is only readjusted, the microscopic fibres of the material are not cut, as done in machining for example, leading to a better texture of the metal and the resulting internal stresses can be capitalised on for the intended application. The important processes in this thesis are the bulk forming process of extrusion and the two sheet forming processes of bending and deep-drawing which are briefly described in the following.

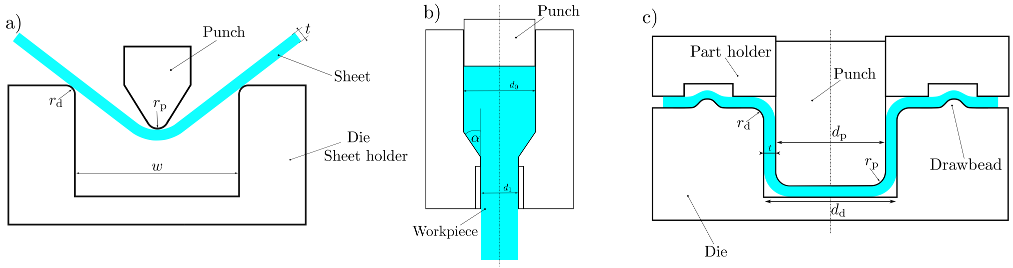


Figure 8.1: The three forming processes of interest in this thesis. a) Free air bending, b) full forward rod extrusion and c) stretch indenting.

Extrusion. The specific extrusion processes of interest in this thesis are full forward rod extrusion and the more complex adaptation of forward hollow extrusion. The former is depicted and annotated in Fig. 8.1b). In this type of forming process, a cylindrical workpiece is reduced from initial cross-section area A_0 to final cross-section area A_1. Since this is a cold forging process, it is undertaken at room temperature. However, due to the large plastic deformations, high changes in temperature arise during forming. This has to be taken into account in the numerical simulations. Due to the rotational symmetry of the process, 2d simulations are possible by exploiting said symmetry. The deformation during forming is achieved by pressing a punch into the die, which in turn pushes the workpiece through the conical forming area of the die. During this transition in geometry, the plastic deformation occurs. After this plastic deformation, and exit of the die, a steady-state can be assumed in the formed lower section of the extrudate. These processes allow creation of a multitude of important components, such as shafts, bolts, nuts, screws and many more, and therefore is an important forming process to consider regarding optimisation.

The significant process parameters which characterise the process are the shoulder opening angle 2α, which guides the material towards its final diameter A_1, and the extrusion strain $\varepsilon_{\mathrm{ex}}$ given by

$$\varepsilon_{\mathrm{ex}} = \ln\left(\frac{A_0}{A_1}\right). \tag{8.1}$$

In the case of full forward rod extrusion, given by the initial diameter d_0 and final diameter d_1 of the workpiece, the above equation can be rewritten as

$$\varepsilon_{\mathrm{ex}} = 2\ln\left(\frac{d_0}{d_1}\right) = 2\ln\left(\frac{r_0}{r_1}\right). \tag{8.2}$$

Application of the same methodology, and including the initial diameter d_{m} of the mandrel, the extrusion strain for the forward hollow extrusion process reads

$$\varepsilon_{\mathrm{ex}} = \ln\left(\frac{d_0^2 - d_{\mathrm{m}}^2}{d_1^2 - d_{\mathrm{m}}^2}\right) = \ln\left(\frac{r_0^2 - r_{\mathrm{m}}^2}{r_1^2 - r_{\mathrm{m}}^2}\right). \tag{8.3}$$

In order to quantify the damage state in an analysed extrusion process, if damage models are not included in the simulation, the well-known stress triaxiality η is used. It is defined as the fraction between the hydrostatic stress σ_{h} and the von Mises equivalent stresses σ_{vM}, i.e.

$$\eta = \frac{\sigma_{\mathrm{h}}}{\sigma_{\mathrm{vM}}}. \tag{8.4}$$

By definition, a positive stress triaxiality coincides with a tension dominated stress state, while a negative stress triaxiality coincides with a compression dominated stress state. A tensile stress state favours damage accumulation, while a compressive stress state shows less damage accumulation. As a result, a negative stress triaxiality is favourable regarding mitigation of damage accumulation. The compressive stresses lead to a closure of the micro defects during forming and negate additional damage evolution during the forming process [94]. Since ductile damage only evolves during the plastic forming section of the extrusion process, lasting from t_0 to t_{end}, the quantifiable measure

$$\eta^{\mathrm{mean}} = \frac{\int_{t_0}^{t_{\mathrm{end}}} \eta(\overline{\varepsilon}^{\mathrm{p}}(t))\, \dot{\overline{\varepsilon}}^{\mathrm{p}}(t)\, \mathrm{d}t}{\int_{t_0}^{t_{\mathrm{end}}} \dot{\overline{\varepsilon}}^{\mathrm{p}}(t)\, \mathrm{d}t} \tag{8.5}$$

is defined. It takes into account the state-dependent load path and triaxiality values of a material point over the course of forming. In (8.5) $\overline{\varepsilon}^{\mathrm{p}}$ denotes the equivalent plastic strain as calculated by the material model for the evaluated material point and $\dot{\overline{\varepsilon}}^{\mathrm{p}}$ is the time increment. The mean triaxiality measure η^{mean} therefore only considers effects caused by plastic deformation and is commonly used to characterise non-proportional load paths by a single scalar value, cf. [47].

Regarding damage in extrusion processes, [221] showed that certain process parameters have significant impact on the resulting damage in an extruded part. Increasing the extrusion strain yields lower triaxiality values, which coincides with a higher compressive stress state in the forming zone. This results in void closure and therefore reduction of damage. While not as significant, a decrease in shoulder opening angle and friction also reduce damage accumulation [111]. Fatigue tests and the Charpy impact energy were used to quantify the simulated results with experimental data. Furthermore, utilisation of scanning electron microscopy (SEM) together with energy dispersive X-ray spectroscope (EDX) measurements in [109, 110] enabled quantification of the accumulated damage on the microscale, and hence further validation of these results. In [90] the damage behaviour in an extruded part was further enhanced without adjustments to the geometry of the formed part. By applying counterpressure to the bottom of the extruded material in varying strength, the triaxiality value in simulations could be drastically reduced. No further changes to the remaining process parameters were required. Examination with SEM measurements of corresponding formed parts showed a significant reduction in pore size and density, confirming the simulated data. For more general information on rod extrusion processes, cf. [135, 139].

Bending. In this work, the well-known process of free air bending, and its adapted form of elastomer bending, are considered for optimisation. The former is depicted in Fig. 8.1a). This relatively simple sheet metal forming process consists of a plate of sheet metal which is placed on top of a sheet holder and then pressed down by the punch. No additional pieces are utilised to keep the sheet in place and only friction between sheet and punch, as well as between the sheet holder and the sheet restrict lateral sheet movement. In similar process variations, such as V-die bending, the sheet metal bottoms out during the process, wherein the punch then presses the sheet into the die. In free air bending, the resulting bending angle in the sheet at the end of the forming process is defined by the punch and die geometry, as well as the amount of displacement prescribed onto the punch. Due to its simplicity, the products of this process are also very straightforward in nature, such as V- or U-profiles.

The process parameters of note in this process are the width between the two sheet holders w, the thickness of the plate t and the radius of the punch and die r_{p} and r_{d}, respectively. The adapted process of elastomer bending, cf. [193, 221], places an elastomer cushion below the sheet. During the forming process the sheet is pressed into this cushion, which leads to reactive compressive pressure onto the bending area of the sheet. Since this area is most critical regarding damage accumulation, cf. [2, 160, 161], a superposition of these compressive stresses enables a closure of voids within the micro structure and therefore improves the damage state of the formed sheet. Hence, similarly to the previous analysis of damage in extrusion, the stress triaxiality η is also a very good indicator for damage in bending, if no damage modelling is performed during simulations. Due to the high tensile stresses in the outer fibre of the bending area, void growth is intensified there as well. This coincides with a high stress triaxiality, cf. [221]. Superposition of the compressive stresses leads to a reduction in triaxiality. For more information on bending processes, cf. [135, 138].

Deep-drawing. The forming process of stretch indenting or cupping is considered in this thesis, see Fig. 8.1c). Similarly to typical deep-drawing, a sheet metal is held fixed between a die and part holder. The latter puts pressure onto the sheet and therefore restricts material flow throughout forming. By lowering the punch into the die and therefore pushing down onto the sheet, the metal takes on the shape of the die. Additional control over the material flow is achieved by the drawbead, which is part of the process and situated on the die. This drawbead restricts the material flow of the sheet metal, which in turn leads to a decrease in material thickness at the bottom of the drawn sheet. Typical products of this process are cups and similar structures.

An important process parameter in deep-drawing processes is the pressure between workpiece and the tools, i.e. the part holder force. If these forces are too high, the material flow is restricted too much which leads to cracks. If they are too low, flange winkling will occur. Additionally, specifically for stretch indenting here, the drawbead geometries impact this behaviour as well. Furthermore, the specific geometries of die and punch are important. The transition radii r_p and r_d govern the material flow at the edges, while the combination of diameters d_p and d_d describe the geometry of the formed sheet. Additional material information often considered in deep-drawing are so-called form limiting curves (FLC), cf. [118]. They describe the formability of the respective sheet material used, and allows the engineer to prevent cracks due to high tensile stresses during forming.

Regarding damage in deep-drawing, the main factors are the die radius and the number of steps to complete the whole forming process. The part holder force does not seem have a significant impact on the damage behaviour. An increase in die radius adjusts the load path in the outer flange, which positively affects the tensile stress state and therefore damage evolution [164, 166, 172]. Production of the same part by using a single-step or multi-step process can vary significantly, depending on the type chosen. A simple multi-step approach has a negative impact on damage evolution, while reverse deep-drawing improves damage behaviour compared to the single-step process [165, 166, 171]. Depending on the type of part which is created within the deep-drawing process, the significant impact of these parameters differs, but the overall resulting conclusions remain.

In this work, the focus is only set onto the geometry of the drawbead to adjust damage behaviour, which simplifies the optimisation. Furthermore, the choice to optimise stretch indenting, instead of typical deep-drawing, is due to the reduction in simulation cost. The non-symmetrical deep-drawing process analysed within TRR188 could only be resolved in a 3d-setting, which significantly increased computational effort. To enable optimisation, the choice of a 2d-problem with stretch indenting was chosen instead, reducing the computational effort and hence enabling optimisation. For more information on deep drawing and related processes, cf. [135, 138].

Regarding failure in forming processes, the fundamental configuration of the parameters for their respective process are generally recognised. Therefore, prevention of chevron cracks in extrusion, cracks within the bending area for free air bending or the prevention of cup base fracture in deep-drawing, see Fig. 8.2, is well understood. However, the leading cause for these macroscopic failure incidents is the accumulation of local, microscopic damage due to the nucleation of voids during forming.

While the criteria to prevent failure are mostly known, their application generally do not consider damage. A better understanding of damage in forming allows improvements on safety margins and better utilisation of the material. This knowledge of how to prevent damage within a given forming process, by altering certain parameters and not changing the final geometry of the part, are still not fully understood. Herein, the concept of numerical optimisation is very beneficial, as it allows the automatic generation of these parameter sets and thereby damage tolerant formed parts.

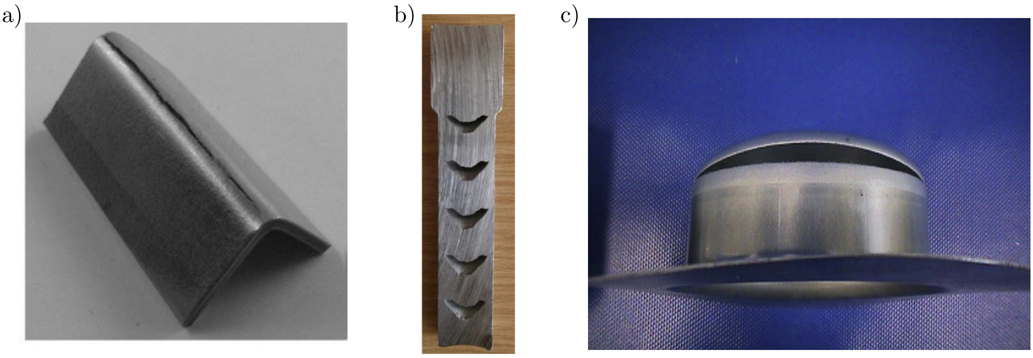


Figure 8.2: Failure in forming due to high accumulation of ductile damage: a) Cracks in the bending area [212], b) Chevron cracks in rod extrusion [211] and c) cup base fracture in deep-drawing [10].

8.3 Choice of optimisation approach

Optimisation of forming processes can be conducted in multiple ways. Two approaches are described in the following. An *indirect* optimisation and a *direct* optimisation.

Within the *indirect* optimisation, the considered process is not simulated as it would be performed in experiments. Instead, certain assumptions are made to allow the performed simulation to be reduced in its complexity. That is, the process is simplified such that it can be modelled and simulated more easily, e.g. by applying boundary conditions such that contact mechanics can be neglected. The mathematical optimisation is consequently applied to the function that arises from this simplified modelling approach. Hence, the optimisation problem can only *indirectly* optimise the reference process due these applied simplifications.

The *direct* optimisation aims to model the forming process with as few simplifications as possible, aiming to closely capture an experimental setup. That is, contact mechanics for example are considered and only simplifications such as symmetry or simplified material models are employed. The boundary conditions are applied according to experimental setups. Hence, the optimisation problem, i.e. objective and constraints, includes process parameters, which directly influence their process simulation and not a simplified replacement model. Therefore, the reference process can be optimised *directly*.

However, the field of numerical optimisation of forming processes in literature itself is difficult to navigate. Often the term "optimisation" is used to describe a general improvement of a considered forming process in a classical engineering sense. That is, no numerical optimisation is explicitly applied. Instead, a process is either improved by experimental data, see e.g. [43, 93], with the approach often called design of experiments (DOE), or initially simulated and afterwards the optimal set of parameters verified with experiments, see e.g. [13, 119]. Sometimes, the experiments are omitted and only simulations computed, see e.g. [75]. In all these cases, the problem is not necessarily described by a mathematical function. Rather, a single or multiple key performance factors, such as punch loads, wall thicknesses, sheet springback, bending angle and more, are manually analysed. While theoretically these types of approaches can be summarised as optimisations with a grid-search method, depending on the choice of sampling points, it will not be included under the possible approaches discussed and considered for the problems at hand. Nonetheless, numerical optimisation, as understood in this thesis, finds application in the field of forming processes.

8.3.1 Indirect optimisation

A beneficial approach to optimise complex processes is to simplify said complexity down to a minimum, while still considering the key deformations and process properties. This simplified modelling approach, which for example aims to exclude contact mechanics, can in turn be used within standard optimisation environments. Consequently, since the process is not depicted in its actual setting, only an *indirect* optimisation of the considered process can be performed. For the forming process of free air bending for example, such a simplification is depicted in Fig. 8.3. Instead of simulating three parts, i.e. the sheet, the die and the punch, and therefore requiring contact mechanics between those parts, only the sheet is modelled. The boundary conditions have to be changed accordingly. One model aims to simulate the process in an inverse manner, clamping the centre of the plate and applying the load on the outer edge. This central punch is replaced by a fixed deformation, which is then applied to the outer edges where the sheet would be in contact with the die, see Fig. 8.3b). The other model exchanges the right part of the die with a fixed bearing and fixes the horizontal displacements on the symmetry axis. The load is applied by a prescribed force or displacement in exchange for the punch, see Fig. 8.3c). This approach is commonly utilised in topology optimisation to simplify three point bending processes, cf. [83, 157, 184].

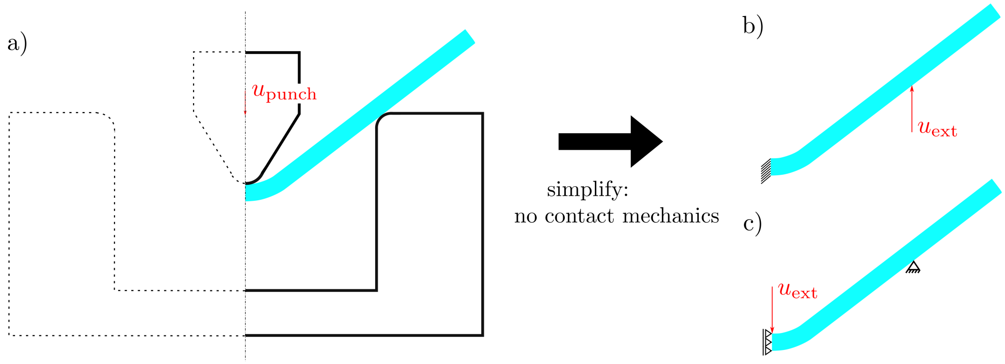


Figure 8.3: Simplification of a bending process. a) The simulation of free air bending (exploiting symmetry), b) and c) the simplified models without contact mechanics. In red the applied displacements.

For simple processes, such as bending, this is straightforward. Model reduction for other forming processes would be more complex, or nigh impossible. Extrusion processes for example are heavily governed by the shape of the extrusion die and thus the resulting contact forces. Therefore, removing this geometry from the simulations and aiming to replace them with fixed boundary conditions, either of Dirichlet or Neumann type, will lead to even larger inaccuracies. While simplification example for bending will introduce slight variations when compared to the accurate simulation, the simplification of extrusion will lead to significantly higher deviations when compared to an accurate model. In addition, friction is completely omitted in such an approach. However, its property might be a relevant process parameter to consider since lubrication is an important aspect in many forming processes. While simulations like these are possible, enhancing this concept further to include optimisation leads to additional difficulties. Since this modelling approach requires application of loads to replace the contact mechanics, it is therefore only possible to alter these external loads and boundary conditions to yield changes in the forming process throughout the optimisation. This further limits the scope of an applied optimisation as the actual forming process and its parameters cannot be accurately optimised.

In such a case, it is more beneficial to utilise load paths from a process and apply these to representative volume elements (RVE), which depict the microstructure of the considered material. Based on the RVE and the load path, both can be optimised to generate optimal micro structures or optimal load paths for the process at hand. However, after optimisation, a link between optimal result and the forming process has to be re-established in order to apply the optimised results to the forming process itself. Therein lies an additional challenge, as an optimised solution not necessarily allows to be generated by the considered forming process. These limitations can be included with constraints into the original optimisation, but this requires a deep understanding of the process. In [61, 62] this is realised for example. Topology optimisation is applied to sheets for deep-drawing by simplifying the forming process. Manufacturing constraints regarding wall thickness of the sheet and prevention of undercuts are considered. The generated geometries with included holes can be formed in one step of deep-drawing. They result in light weight structures wherein the holes are introduced into a geometry before forming which are optimised. Due to considered constraints in the optimisation, these are able to withstand the forming process. Since the holes are inserted into the flat sheet before forming, this circumvents the complex adjustment of topologies for the formed sheets without the additional perforations. However, they note that certain process parameters such as drawbead or punch geometries have to be omitted in such an optimisation, because the simplification does not allow their consideration.

Another approach in sheet metal forming to avoid contact mechanics is the inverse approach, wherein the final and initial deformation are known beforehand, and the steps in between are computed based on this knowledge, cf. [20, 59, 168]. These approaches allow modelling of the considered process without contact, however require precise knowledge of the boundary conditions which have to be applied.

Another approach to simplify a forming process, is to apply numerical optimisation to analytical functions. That is, the processes or specific problems at hand are described with simplified functions, based on analytical and geometrical properties. These are subsequently minimised without the need for a FE simulation. In [88], the blank holder force within deep-drawing is optimised by specifying it depending on variables, such as the bending angle or other geometric variables of the process. The analytically derived quantity which describes the process, is optimised with numerical optimisation, while simultaneously subjecting it to constraints which prevent tearing and wrinkling.

A final alternative would be the use of surrogate models. These can allow quick computations of complex forming processes and do not require computation of contact mechanics. One such approach, often used in the experimental field of optimisation of forming processes is the response surface methodology (RSM), cf. [39, 104]. This approach fits a polynomial over the conducted DOE with the process variables as inputs. The output describes the desired objective of the optimisation which can be numerically optimised, cf. [7, 67, 227, 231] for application in forming.

While these presented approaches are applicable, each have their drawbacks when multiple different forming processes have to be considered simultaneously. Many of the presented approaches require a deep understanding and long research while only considering one forming process. However, in the context of TRR188, multiple forming processes need to be considered and the derived approach has to be applicable for all respective processes. Due to the complexity and the huge differences between these considered processes, assumption regarding simplification and their *indirect* optimisation cannot be directly transferred, leading to the next, *direct* optimisation approach described in the following.

8.3.2 Direct optimisation

The *direct* approach aims to use a given forming process, which accurately models the physical process, and optimises it directly. That is, the process parameters $\mathbf{s}_\mathrm{p}$ which describe the process in question are the design variables $\mathbf{s}$ of the process and altered during the iterative optimisation. In contrast to the previously described approach, i.e. *indirect* optimisation, this allows direct adjustments of these parameters and therefore the actual process. Consequently, strict control over certain production-specific constrains that need to be considered is possible. In addition, since many different forming processes are considered and optimised, the application of a *direct* optimisation method yields the benefit to easily and swiftly exchange working simulation models. Furthermore, the main application for this optimisation framework is the optimisation of damage within a given process. This property can be optimised due to many different factors within a forming process. These consist of geometric type, such as radii and die geometries, boundary conditions, such as external forces or displacements, or the lubrication, i.e. the friction parameter within the simulation. Therefore, the utilised optimisation framework requires consideration of many different kinds of design variables simultaneously. Only restricting itself to one type of optimisation, e.g. structural optimisation, is not feasible in this case. Since the forming processes are directly optimised, the contact mechanics need to be included in this approach as well. Therefore, an environment is required, which enables simulation of contact problems. Consequently, the choice of solution strategy is very critical. Since contact algorithms introduce a discontinuity into the mathematical formulation, the application of sensitivity information due to numerical gradients is influenced. A general application of gradient-based methods is therefore not advisable and slow gradient-free methods need to be considered as well.

Additionally, this approach comes with other challenges that have to be considered, and are addressed in the following chapter, see Chap. 9. Most importantly, the contact mechanics require a robust framework which can handle the requirements, such as friction effects and multi-body configurations. Additionally, simulation of the complete forming process takes a lot of computational effort which, combined with the lengthy optimisation, leads to very long computation times. Another problem that can occur is convergence problems within the FE simulations for atypical combinations of process parameters. The optimisation algorithm can yield, or momentarily iterate into, parameter combinations which can lead to unsuitable steps, which can prevent the FE simulation from converging. This has to be considered in the algorithm. Either by throwing a high objective value in this iteration spot, and therefore diverge the optimiser from this unsuitable location in the design space, or using many starting guesses for the optimisation, to approach certain design spaces from multiple angles and circumvent undesirable parameter combinations.

This application of *direct* optimisation is also applied in literature. In [121] for example, deep-drawing is numerically optimised by optimising blank holder and drawbead forces. They also mention the complications regarding quantification of sensitivity data for problems with contact mechanics. In turn, they choose response surface methodology to enable optimisation without gradient-based methods. Other applications of *direct* optimisation applied to forming process are presented in e.g. [131, 133, 223].

Chapter 9

Optimisation using commercial software (Abaqus)

This chapter describes the developed optimisation framework IOPS using the commercial finite element software Abaqus. The choice for this software is motivated. The environment for IOPS is briefly described and files and nomenclature listed, to ease the reader into the presented optimisation framework. IOPS allows incorporation of already existing simulation models, therefore the necessary adjustments for these is emphasised. Possible approaches to structure an optimisation framework around Abaqus are outlined, highlighting the benefit of the chosen approach ofr IOPS. Its general applicability is tested with some benchmark examples.

9.1 Requirements for the computational environment

In this chapter the additional optimisation framework IOPS is described. It enables a general optimisation of arbitrary problems, with the focus on forming processes. As motivated in Chap. 8, a framework is required, which allows optimisation of forming processes, and therefore optimisation of nonsmooth problems. Specifically, contact mechanics have to be considered. While approaches exist, which circumvent the contact simulations, they were discarded due to certain drawbacks, see Sect. 8.3. In the context of *direct* optimisation, this puts two main constraints on the choice of commercial finite element (FE) software at the core of the optimisation framework:

1. A robust implementation of contact mechanics and,
2. a framework which enables external access to modelling and simulation data.

The first requirement can be easily met in many different commercial software. Contact mechanics are an established field in numerical continuum mechanics and therefore implemented in many FE environments, e.g. Abaqus, Ansys, Comsol, FEAP and many more [9, 56, 58, 220]. The second criteria is crucial to provide a gateway for the numerical optimisation. In order to automatically generate a new design, the iterative data has to be changed within the simulation without manually accessing it each time. Therefore, an interface has to be provided by the software to enable automatic design changes.

Based on these criteria, the decision was made to utilise the commercial software Abaqus [58], around which the optimisation framework was structured. This software is specifically chosen for three main reasons:

1. It is widely used within TRR188, see Sect. 8.1, allowing easy cooperation with different projects and the exchange of simulation and material models,
2. it provides a robust implementation of contact mechanics and includes the material models necessary for the simulation of the considered forming processes and,
3. it allows easy automation via scripting in Python and thereby a good basis for an optimisation framework.

The concept of using commercial software to simulate forming processes and structuring an optimisation framework around said commercial software has been discussed in literature before. In [46] deep-drawing is simulated in Abaqus and optimised based on the simulated results. Based on the geometrical description for the sheet deformation, a minimisation of required applied forces is defined as the objective function. To reduce the computation time, an inverse approach for the simulation is chosen. In [209] the springback for U- and V-shaped bending geometries is optimised. Therein, using the Python interface, Abaqus is coupled to an genetic optimisation algorithm written in Fortran. Recently in [106], vaporizing foil actuator forming was optimised by using Abaqus as the solver. By defining parameters which govern the pressure distribution as the design variables, optimal charge setups could be generated to form the desired sheet shapes. The optimisation problem was solved using Isight with a NSAGA-II algorithm. The optimised results were afterwards validated in experiments.

Regarding optimisation with Abaqus outside the application of process optimisation, [197, 198] presented a framework around Abaqus to solve a number of parameter identification (PI) problems. Since the framework only focuses on PI and, the design variables only take material model parameters into account. These can either be altered in the Fortran code of said utilised material models or directly in the input files of the simulations. The general concept finds further application in other fields of research, cf. [6, 240, 243].

However, all these approaches only consider one type of problem at hand and therefore do not necessarily allow application to other type of problems nor consideration of different type of design variables simultaneously. This is however required for the problems of interest in this work and have to be accounted for. IOPS therefore has a modular structure, allowing the user to prescribe objectives and constraints freely. It is consequently designed to take any Abaqus simulation model and enables optimisation of many different problems without being constrained to one type of optimisation problem. Furthermore, many different types of design variables can be chosen simultaneously.

Due to its modularity, adjustments outside the FE environment do not restrict themselves to optimisation. In [30] for example, the Python scripts were used to simulate wear on machining parts. Therein, the scripts are utilised to erase critical elements within a sheet-bulk metal forming process due to wear on surfaces, resulting from steady friction between the bodies in contact. The script detects critical elements in a post-processing step and alters the surface of the tool for the next simulation model based on gathered simulation data from the previous iteration.

9.2 Optimisation framework

In this section, the optimisation framework using Abaqus is described in detail. Even though the presented framework is derived to enable damage optimisation of forming processes within TRR188, its structure does not limit itself to these kinds of problems. The choice of design, objective and constrains is freely definable and up to the choice made by the user. An additional benefit, in contrast to pure structural optimisation (SO), is the multi-design optimisation possible in this environment. Multiple types of problems can be optimised simultaneously since the modelling is provided within the FE software. This enables SO together with other types of optimisations, e.g. boundary conditions, simultaneously. For an overview regarding possible design variables, objectives and constraints, see Table 9.1. These benefits are further discussed in the benchmark examples in Sect. 9.3.

Table 9.1: An exemplary overview regarding possible optimisations with the described optimisation framework.

optimisation type	design variables	examples
geometric	control points (CAGD), part dimensions, nodal coordinates	$\mathbf{X}$,$\mathbf{p}$, r, w
parameter identification	constitutive/material parameters	$\mathbf{m}$,E,ν,σ_{y0}
displacements	displacements, strain	$\mathbf{u}$,$\boldsymbol{\varepsilon}$
loads	forces, stress	F,$\boldsymbol{\sigma}$
interactions/lubrication	friction parameter	μ
load paths	amplitude (boundary conditions)	$\mathbf{s}$

9.2.1 Commercial simulation software Abaqus

Abaqus [58] is a commercial FE software, able to compute many different FE problems. However, in this thesis only the implicit solver Abaqus/Standard for the quasi-static cases, and the explicit solver Abaqus/Explicit for dynamic problems with complex contact interactions is taken into account. The software has many inbuilt material models. However, users are able to provide additional material descriptions with user materials UMAT. Initially, these had to be provided in Fortran, but nowadays C and C++ implementations are also possible. The simulation call has to be adjusted accordingly, e.g. for user files with the addition of "user=UMAT.for".Since Abaqus only provides displacements and temperature fields, inclusion of nonlocal damage problems requires definition of user elements UEL. However, these have significant drawbacks, such as loss of parallelisation, contact mechanics, visualisation and more [230]. For nonlocal damage, [177] presented a framework which exploits the heat equation and therefore the temperature field to regularise damage. This allows utilisation of the previously lost features due to the user element, while still incorporating nonlocal quantities.

To interact with the software, a user generally uses the provided graphical user interface (GUI) of Abaqus/CAE. This interface allows a user to create, view and adjust simulation models for a given problem. In addition, Abaqus/CAE is used to analyse the simulation data within Abaqus/Viewer. Most importantly regarding automatisation, Abaqus/CAE can also be prompted to start without the GUI with the "noGUI" parameter. This enables modification via scripting and therefore is a key aspect when considering application of scripts within an optimisation framework. These prompts, similarly to running simulations, can be achieved by command prompts within a command window.

A brief overview over important files and names regarding Abaqus are given, as they are used within the next section to describe the implementation of the optimisation framework in detail.

Abaqus model database `.cae`: This file contains all necessary information of the simulation model. Generally, this file is accessed only through the GUI to apply changes to the simulation model. From this model database the input file can be generated to run the FE simulation. However, scripting with Python also enables changes to this type of file and its stored data. This data can only be interpreted by using Abaqus/CAE.

Analysis input file `.inp`: The input file necessary to run simulations with the chosen FE solver of Abaqus. The data is of plain text and therefore readable and hence directly interpretable. Simulations can be run with only this file, without the original model data base being required. This file is created by the model database and stores all necessary data like the mesh, connectivities, boundary conditions, sets etc.

Abaqus replay file `.rpy`: The replay file captures all inputs a user performs while interacting with Abaqus/CAE. The commands stored within this file are stored as plain text Python prompts. Running this file within Abaqus/CAE will replay the inputs the user made, including adjustments of view space and camera. Only the five most recent files are automatically stored within the work directory. Since these files are not associated with a specific model, but with a session, their content is either incomplete, or a file is straight out automatically replaced.

Abaqus journal file `.jnl`: Similarly to the replay file, the journal file stores information regarding model adjustments made by a user. This file is automatically saved together with the model data base and therefore associated to this `.cae`. In contrast to the replay file, not all inputs the user made are stored. For example, camera and view port changes are not recorded. The main application for the journal file is the recreation of a model due to unforeseen crashes of the software. Since the input in the journal file are stored in plain text Python code, the content is very valuable when considering scripting within the context of optimisation.

Output database `.odb`: The output database stores the simulation data for the computed FE simulations. They get created during the simulation and updated after each converged global iteration. Therefore, even if the complete simulation does not converge, iterative data can still be accessed and analysed. The access can either be achieved within the GUI or by loading the data with Python scripting. The latter access is very important, as it allows the specific analysis of critical measurements for the optimisation. Both require interpretation through Abaqus/CAE, similar to the `.cae`.

Part: Parts are components of a simulation model which are to be simulated. They are defined individually and combined in the assembly. Material properties have to be chosen and they have to be meshed with a desired element type. The boundary conditions have to be applied to each considered part as well.

Sketch: The model sketch is the Computer Aided Geometric Design (CAGD) model of a created part. In addition to creating the part, it can be annotated with dimensions. Changing these dimensions will adjust the model and therefore the part accordingly.

Partition: Partitions are areas within a sketch of a part, where it is separated into smaller regions. The model is still of one piece from a numerics point of view. For example, use of partitions allows application of different material properties at different locations of a given part, e.g. changes in stiffness in a local material model, or changes to allow finer meshes for important sections of a part. Furthermore, utilisation of partitions allows fine adjustments regarding specific definition of sets.

Sets: Set are an assembly of nodes, elements or geometry, which the simulation software can use to locally apply conditions or store data. Utilisation of sets allows specific definition of areas where quantities shall be evaluated, or in the context of optimisation optimised or constrained.

9.2.2 Optimisation process

The general optimisation sequence is specified in Fig. 9.1. To solve a defined mathematical optimisation problem, the framework can either employ gradient-based strategies in the Matlab function *fmincon* or the gradient-free method of *fminsearch*. The latter generally results in overall better solutions at the added draw-back of a higher computational effort. The choice is up to the user by selecting the desired method in the pre-processing step. For the gradient-based optimisation, the function can apply different methods, e.g. Sequential Quadratic Programming, the Interior-Point Method and more. For an overview and further options, see the detailed documentation [156]. The gradients have to be computed numerically, i.e. by means of the Finite Difference Method (FDM). This is due to the necessity of contact formulations to accurately model forming processes, see Chap. 8. With the choice of Abaqus [58] as the simulation software for the FE problems, it has to be inserted into an optimisation procedure. For the specific chosen approach emphasised in the following, these adjustments regarding incorporation of design changes are highlighted in red within Fig. 9.1.

In turn, utilisation of numerical gradients has significant drawbacks within the application of said contact algorithms. Since contact problems are inherently discontinuous, a numerical gradient by means of the FDM will therefore be derived with significant errors. This topic of noise within the application of the FDM is discussed in [174]. These gradients may lead to insufficient or wrong search directions for the optimiser and in turn to bad or out-right wrong solutions. While this may not always be the case, like in [97, 101], where contact interactions were considered and still suitable results generated, it should be taken into account when choosing an optimisation solver. Therefore, it is generally advisable to use the gradient-free method for problems with many contact interactions, as they are discussed in the following chapter.

The gradient-free method utilised in the Matlab function *fminsearch* is a Nelder-Mead simplex, see Sect. 3.3.1. As was discussed, said algorithm can generally not handle constraints. To treat constraints nonetheless, the penalty method is applied in two steps. Constraints can be generalised into two groups. The first type of constraints is simulation dependent, i.e. the FE simulation has to be conducted until they can be evaluated, e.g.

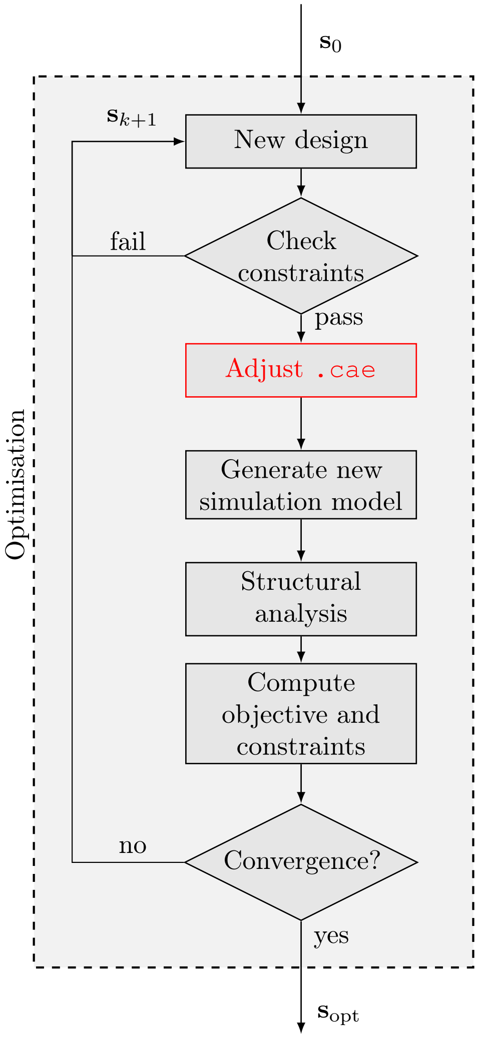


Figure 9.1: Optimisation flow chart. In red, the use of Abaqus/CAE to update the model.

stresses, strains or damage variables. These constraints can be included by adapting the evaluated objective function according to (3.7).

The second group of constraints are simulation independent and can be evaluated before running the FE simulation. These types of constraints can be for example of geometric type and evaluated beforehand without the need of simulation data. Since the objective function is unknown at this point and therefore cannot be directly influenced, instead a high function value, i.e. `Inf`, is returned. This forces the optimiser to treat the current iteration as unsuitable and to try a different step for the current iteration. The approach is taken from [57], which has been used in this framework.

If those constraints are fulfilled however, a new simulation model can be generated. This in turn creates a new input file, which is subsequently called to run the FE simulation. After computing the new structural analysis with design variables $\boldsymbol{s}_k$ of iteration k, the objective function and additional, non-trivially evaluated constraints are computed. This process is conducted until a predefined break threshold is met, which in turn means convergence of the optimisation. Throughout the optimisation the data is stored for the post-processing routine, see Fig. 9.2.

9.2.3 Optimisation framework with Abaqus

The main problem when dealing with optimisation utilising commercial software is the choice of how a newly generated design is incorporated and adjusted within the underlying simulation model. In this case, using the commercial FE software Abaqus [58], there are multiple approaches when applying the numerical generated changes in design to the FE model. The most straightforward, here denoted as the *IPF* approach, is adjusting the necessary variables in the input files. These files are submitted to the FE solver to run the job and generate the subsequent results. They can be used to change simple design problems, e.g. forces and amplitudes like in [97, 101] or changing material parameters to conduct PI, cf. [198]. Initially, this was the approach that was followed [95, 97, 101]. Altering the input file is easy to accomplish and does not require a lot of insight regarding internal definitions within Abaqus, like how model data is stored and where to access it. Additionally, this approach is easier to access, since the data which is adjusted is stored in plain text format and therefore errors are easy to identify.

However, when dealing with geometric design optimisation, the process of altering the design can become more complicated using this approach. The mesh outside the Abaqus/CAE environment must either coincide with the original mesh or a mapping is required to transfer the generated mesh by the optimiser to the mesh used in the FE calculation itself, cf. [95]. The overall main drawback with this approach, i.e. changing the design in the input file, is the required additional definition of said design outside of the actual Abaqus/CAE model, which is prone to errors when more complex problems are chosen to be optimised.

To enable optimisation of said more complex geometries, e.g. the geometry of an extrusion die with variable radii, the framework must be able to handle the aforementioned problematic design definitions. One approach, denoted as the *RPY* approach, could be to use the Python commands which Abaqus saves in the journal and replay-files. While this may solve the above problem, certain commands executed within the Abaqus/CAE GUI are not executable outside the GUI. Consequently, these cannot be used when dealing with the automated process of numerical optimisation. Furthermore, since these replay-files are not necessarily manually kept track of and saved, they may get overwritten or deleted when the GUI is closed and reopened. This is even more difficult due to the desired cooperation with other projects of TRR188. As generally these files are not accounted for by the user, they are generally not saved and stored for later use.

The final framework of this thesis instead is to directly change necessary parameters within the model data base for the simulation, denoted as the *MDB* approach. Changes within this database, and respectively newly generated input files, allow for an automated framework for the applied numerical optimisation. While this approach is very effective, it requires a certain understanding of how the model database is structured and how to change certain values within. For this purpose, the reader is directed to the thorough documentation [58].

In theory, application of the *MDB* approach in IOPS allows for any kind of definition for the choice of design variables within the optimisation framework, e.g. geometric design, material parameters, amplitudes, boundary conditions, general simulation parameters etc., and thereby depends on the chosen problem, see Table 9.1. In order to change these parameters, they have to be parametrised within the model to be changed later on. For geometric process parameters for example, this is accomplished by defining these parameters as dimensions of the model sketch, which can be changed by means of Python scripting outside the GUI. The complete concept of this optimisation framework is summarised in Fig. 9.2, stating the necessary software utilised in each respective step explicitly. The pre-processing, described in Sect. 9.2.4, mainly requires the Abaqus/CAE environment and Python coding, with initialisation of the optimisation problem in Matlab. The optimisation, see Sect. 9.2.5, is called in Matlab which in return calls Abaqus and the necessary Python scripts. The post-processing, see Sect. 9.2.6, revolves around storing the iteration data and additional visualisation data in Matlab. The framework is structured such that all necessary steps are applied in the pre-processing step. The optimisation can subsequently be conducted without any further input from the user.

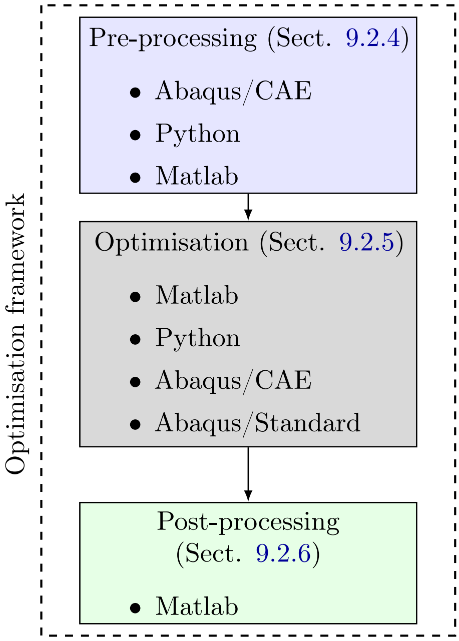


Figure 9.2: General concept of the complete optimisation framework and the software utilised within each step.

9.2.4 Pre-processing

The flow chart for the pre-processing step is depicted in Fig. 9.3. Initially, the FE model for the simulation has to be defined in Abaqus/CAE. In addition to the definition of the general FE model, e.g. parts, mesh or boundary conditions, the underlying design variables have to be parametrised such that they can be changed with dedicated Python code. Additionally, sets have to be defined where the objective function and possible constraints are evaluated. These sets coincide with FE nodes and elements, which in turn store data like stress, strains or displacements. They can be of geometric, elemental or nodal type and are used within the Python script to ensure the correct evaluation of objective function and possible constraints. Even though the type of set can be chosen arbitrarily, the choice of geometric type is advantageous since elemental and nodal sets may no longer be valid if the mesh of their respective model part changes throughout the optimisation. This can for example happen, if the thickness of a part changes, resulting in a necessary remeshing of said part, which in return removes the elements or nodes from a previously defined set. Choosing to define said geometry by means of a geometric set still yields usable data even after remeshing, since it is not coupled to a fixed FE mesh. For this purpose, partitions may be necessary to reduce the size of analysed data. This however has the disadvantage that the user must ensure that the evaluated data remains comparable, i.e. an element of the defined geometric set has the same respective orientation/position in the part even after a change in geometry occurs.

After defining the sets, two Python scripts have to be created:

1. The first script is used to adjust the changes in the `.cae` model data base, wherein the newly generated design variables are used as input variables. It is applied to the Abaqus model database within the optimisation and is called before each simulation during subsequent optimisation steps.
2. The second script is required to evaluate the data stored in the output database. It evaluates the results, and outputs the data for the objective function and possible constraints. It is called after each FE simulation.

The framework is set up in such a way that the problem can be defined modularly, i.e. the required Python scripts can be defined for a variety of optimisation problems. This way, the necessary Python scripts can be specified in the problem definition to incorporate process-oriented output routines and model data base adjusting scripts. As a result, the optimisation framework is not inherently restricted to geometric problems. Additionally, it allows for specific inputs, if further prompts are necessary in a given simulation like user fields or user materials. Finally, the Matlab problem is initialised by prescribing all required data for the optimisation and the optimisation is initiated.

9.2.5 Remarks on the optimisation procedure

The overall optimisation framework was already discussed in Sect. 9.2.2, which follows the flow diagram as depicted in Fig. 9.1. Of importance, with regards to the implementation at hand, is the red coloured box. At that point during the optimisation the Python script to change the design variables within the Abaqus model database is called. This only occurs after the simulation independent constraints are fulfilled, see Sect. 9.2.2, ensuring

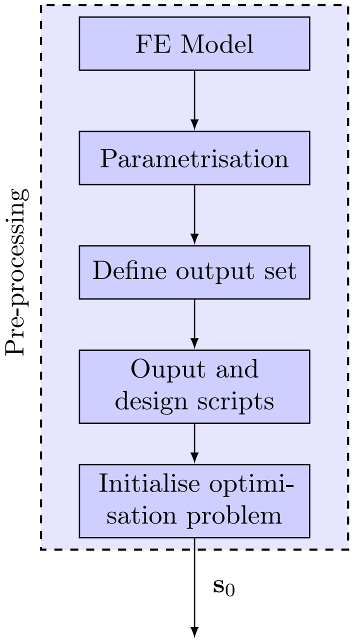


Figure 9.3: Pre-processing flow chart of the optimisation framework.

creation of a valid simulation model. The second required Python script is called after each FE simulation. The data created from this is subsequently used to evaluate the objective function and additional, simulation dependent constraints. Certain variables from these simulations, as defined by the user in the pre-processing steps, are also stored for use in the post-processing step.

To easily analyse results and transfer data between Matlab and Python, the openly accessible libraries *NumPy* [107] and *SciPy* [228] for Python are used. The former allows structuring of matrices and vectors and enhances Python with additional mathematical basics, while the latter is mainly used to store the evaluated data in `.mat`-files. These files can be directly imported in Matlab without the need of reading `.txt`-files or similar data. Similarly, the iterative results from Matlab are stored in `.mat`-files to be subsequently loaded using *SciPy* to adjust Abaqus model data.

9.2.6 Post-processing

The post-processing utilises the evaluated data from the simulations to generate output data to visualise each optimisation step. Therefore, iterative data like objective function and constraints are saved for later use. Additionally, `.vtk`-files are generated within each iteration step to visualise the geometric changes, as well as the changes of field and state variables, for a prescribed process afterwards. The data to be stored has to be predefined within the pre-processing step and needs to be evaluated in the corresponding Python scripts. This steps uses the same routine as AOPS and allows visualisation with Paraview.

9.3 Numerical examples

To present the applicability of the framework for forming processes and other types of problems, some benchmark problems are presented. The process of free air bending is chosen, as it is rather simple as far as forming processes are considered. The first example is structured very simply to present some preliminary results. The second problems aims to emphasise the possibilities by choosing many design variables of different type and optimisation thereof.

9.3.1 Simulation model

The FE model for this simulation consists of three parts: a deformable sheet plate, a punch which pushes into the sheet and, together with the die, bends the sheet into the desired shape. Contact of master-slave type between punch and sheet, as well as die and sheet are defined, with Coulomb friction and a coefficient of $\mu = 0.3$. The punch and die are modelled as rigid bodies. Hence, only the sheet is assumed to be deformable, with the material parameters for DP800 being provided by IUL, TU Dortmund University and taken from tensile tests for the chosen material. An elasto-plastic material model is used to simulate the material behaviour of the sheet. A 2d case is assumed, resulting in a plane strain case for the simulation. Exploiting symmetry conditions allows modelling of only half of the complete process, see Fig. D.2, using 2504 elements of type CPE4 to model the sheet. The mesh is not uniform, instead it is meshed finer at the centre where contact mechanics are crucial and coarser towards the right boundary. The width between the two sheet holders of the die is equal to $w = 24\,\text{mm}$ with the metal sheet having a thickness of $t = 1.2\,\text{mm}$. The punch is pressed into the sheet until a prescribed displacement of $u_{\text{punch}} = 5\,\text{mm}$ is achieved. The punch as well as the die have a radius of $r_{\text{p}} = r_{\text{d}} = 1\,\text{mm}$.

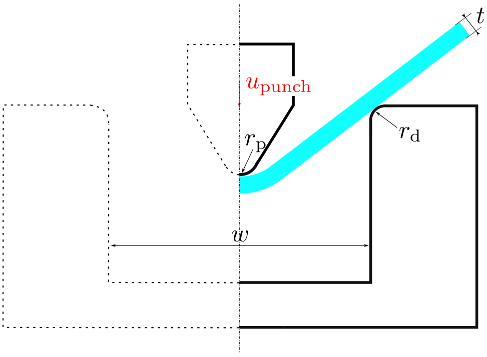


Figure 9.4: Sketch of the simulation for the bending optimisations. By exploiting the symmetry of the process, the dashed part is not modelled.

9.3.2 Optimisation of deformation loads in bending

For the first optimisation problem, the bending process is to be recreated from no prior knowledge on how much the punch is displaced. Hence, the optimisation problem reads

$$\underset{s_\mathrm{l} \leq s \leq s_\mathrm{u}}{\text{minimise}} \quad \mathfrak{J}^\mathrm{b} = \|\mathbf{u}(s) - \mathbf{u}_\mathrm{pre}\|^2, \tag{9.1}$$

with the design variable is only the punch displacement, i.e. $s = u_\mathrm{punch}$. The objective of the optimisation is to recreate a deformation which is prescribed a priori. This displacement field is taken from a simulation with the same boundary conditions as described in Sect. 9.3.1 and compared to the results from the simulations in each optimisation step. Only the surface nodes of the sheet are taken into account; the inner mesh is not considered. Since the prescribed deformation is created with the same model, in theory the optimiser should be able to find the exact solution for u_punch which yields $\mathbf{u} = \mathbf{u}_\mathrm{pre}$.

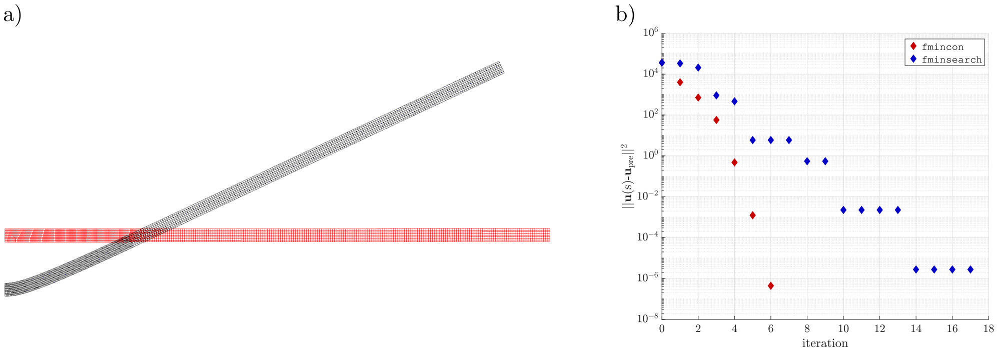


Figure 9.5: a) The initial (red) and optimal or prescribed (black) deformed sheets for both solution strategies. b) The objective over iterations.

The problem is solved with the gradient-based optimiser *fmincon* and the gradient-free *fminsearch*. The gradients for the gradient-based solver are provided by means of the FDM, see Sect. 3.4.1. Since the contact interactions in this forming process are not as pronounced as for example in extrusion, the gradient-based methodology is able to yield a converging solution. The results are depicted in Fig. 9.5. The initial guess for both optimisation problem is set to $u_\mathrm{punch,init} = 0.01\,\mathrm{mm}$, resulting in the initial red, almost flat sheet in Fig. 9.5. Both solution strategies are able to almost exactly reproduce the result, i.e. $u^\mathrm{fmc}_\mathrm{punch,opt} = 4.999\,\mathrm{mm}$ and $u^\mathrm{fms}_\mathrm{punch,opt} = 5.001\,\mathrm{mm}$, where the difference stems from the choice of the break threshold. As expected, both results almost coincide with the black prescribed mesh. Therefore, no difference in the mesh can be observed and therefore only the black prescribed mesh is depicted in the figure. Most notably, the gradient-free method finds a solution in fewer iterations, even though gradients for a nonsmooth problem are considered, see Fig. 9.5b).

9.3.3 Optimisation of springback in bending

A well-known effect in bending is the elastic springback due to the elastic relaxation of the sheet, which happens when it is unloaded by retracting the punch. Due to plastic deformation during bending, the resulting plastic strains yield the final deformation after unloading. However, this does not coincide with the total deformation at maximum displacement u_{punch} during the bending procedure. Hence, the optimisation problem in this exemplary benchmark aims to consider the springback property and optimise the forming simulation, such that the final deformation after unloading is predicted.

The new optimisation problem reads

$$\underset{\mathbf{s}_{\mathrm{l}}\leq\mathbf{s}\leq\mathbf{s}_{\mathrm{u}}}{\mathrm{minimise}} \quad \mathfrak{J}^{\mathrm{sb}} = \|\mathbf{u}_{\mathrm{un}}(\mathbf{s}) - \mathbf{u}_{\mathrm{pre}}\|^2, \tag{9.2}$$

where the prescribed deformations are the same as in the previous section, i.e. before unloading. The deformation $\mathbf{u}_{\mathrm{un}}$, which are compared to the deformation of the surface nodes, after the punch is retracted, i.e. the deformation in the sheet is only due to prior plastic deformation. The design variables in this problem now consider geometric parameters of the punch radius r_{p} and half die width w_{h}, the boundary condition for the prescribed punch displacement u_{punch}, and the friction parameter μ. Therefore, the set of design variables reads $\mathbf{s} = \{r_{\mathrm{p}},w_{\mathrm{h}},u_{\mathrm{punch}},\mu\}$. In contrast to the previous problem, here the set of parameters which can create this deformation is not known, hence an exact recreation of the deformation field is not guaranteed.

Table 9.2: Overview of the springback optimisation.

$\mathbf{s}$	$\mathbf{s}_{\mathrm{l}}$	$\mathbf{s}_{\mathrm{u}}$	initial	optimum *fmincon*	optimum *fminsearch*	optimum *fmincon* weighted	optimum *fminsearch* weighted
r_{p}/mm	0.5	3	1	2.6154	1.1169	1.9167	2.9980
w_{h}/mm	8	20	12	12.0244	12.0201	12.2336	12.2276
u_{punch}/mm	4	8	5	6.2579	6.2685	6.1807	6.1724
μ	0.05	0.6	0.3	0.2185	0.2587	0.0661	0.3379

The initial shape of the sheet after unloading, and hence the differences due to the springback, is depicted in Fig. 9.6a). The black portion of the mesh therein depicts the prescribed deformation of the sheet, which is the result of the forming process without unloading and identical to the previous section. An overview regarding the design variables, their bounds, initial and optimal values are summarised in Table 9.2. The solution to the optimisation problem described in (9.2) is depicted in Fig. 9.7a). The optimiser is able to fit the mesh remarkably well. The sheet after unloading almost coincides with the prescribed deformation. However, a small difference of about the height of 1 element still is visible for both optimisation strategies at the left-most edge of the sheet. As expected, the punch has to be displaced more to achieve this result. However, the other process parameters seem to not affect the outcome significantly more, as both solution strategies yield quite a variance in results. For example, the gradient-based solution for the punch geometry more than doubles the radius, while the gradient-free solution only sees a small increase. Both solutions however slightly increase the die width.

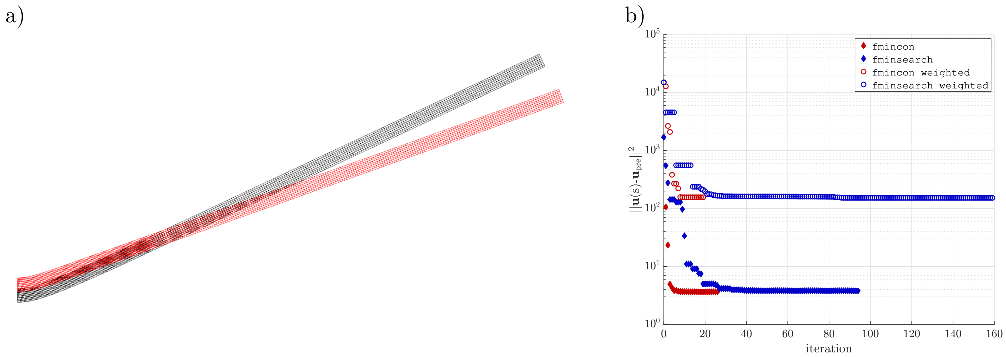


Figure 9.6: a) The initial deformed sheet after unloading in red and and the prescribed deformed sheet before unloading in black. b) The objective over iterations for both solution strategies and both objectives.

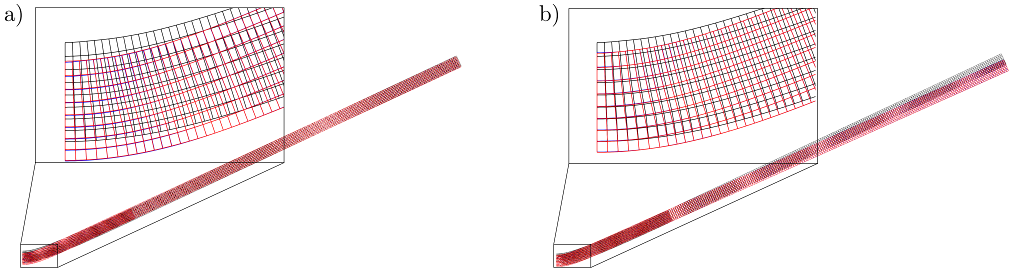


Figure 9.7: a) The optimal state after unloading for the springback optimisation. b) The adjusted problem with the weighted deformation data. The prescribed deformation in black. In red, the solution from *fmincon*; in blue the solution from *fminsearch*.

A drawback of this optimisation problem however, is that all nodes are equally weighted. Due to the lever length, small adjustments for the design variables have a higher impact on the mesh on the right side of the sheet. Therefore, the fit there almost matches the prescribed deformation exactly. However, an accurate match for the central part might be more desired. For this reason optimisation problem (9.2) is slightly adjusted by weighting the difference of the displacement data for all nodes on the left side of the die by a factor of 10. This weighted and adjusted optimisation problem aims to improve the matching of deformation in the centre of the sheet, knowing that the agreement towards the right edge will drop significantly. The results for this optimisation are depicted in Fig. 9.7b) and shows that a better agreement can be found as desired, but with the expected drawback. The latter can be also observed in the iterative graph, see Fig. 9.6b), as the objective value is 100 times higher than the previous result due to the added weights for the left side and hence divergence on the right side of the sheet. To reach this adjusted result, the punch displacement is slightly reduced and the die width slightly increased, see Fig. 9.2.

To further enhance this optimisation problem, one could for example add further parameters such as the die radius or additional forces which could then stand-in for additional work tools. But since this only aims to present an example for the application of the presented optimisation framework, this not further pursued. Nonetheless, the data and results show the applicability of this framework to complex problems with a wide degree of different design variables. Furthermore, gradient-based methods can be applicable if the underlying contact interactions are not severely impacting numerically derived gradients.

Chapter 10

Damage controlled forming processes

In this chapter, results of damage controlled process optimisation for three different forming processes are presented. The results are generated by the optimisation framework IOPS described in Chap. 9. To initially validate the accuracy of the applied optimisation framework, it is applied to full forward rod extrusion and the results compared to ones gathered by normal engineering methods of optimisation. Afterwards, forward hollow extrusion is optimised with two, in complexity increasing, optimisation problems. In bending, an optimal setup for the cushion in elastomer bending is generated by means of a shape optimisation approach. The last forming process optimised is stretch indenting. The process is adjusted by changing the drawbead dimensions. Based on a multitude of initial guesses and subsequent optimisations with two different optimisation problems, a general conclusion regarding the drawbead geometry is drawn.

10.1 Numerical damage optimisation applied to forming

In Chap. 9, a framework for the general optimisation of problems with the commercial software Abaqus has been presented. Its creation is motivated by the necessity of optimising problems with contact mechanics. While the framework itself is not limited to such problems, see Sect. 9.3, its main application is the field of forming processes.

Engineering type analysis, i.e. parameter study, of forming processes with respect to their damage behaviour is very time intensive. The chosen process parameters have to be manually defined and the simulations adapted accordingly, cf. [111, 161, 172]. Applying numerical optimisation allows the automatic generation of process parameters sets, without further input beyond the setup of the optimisation. In addition, the choice of the analysed process parameter sets is always subjected to some form of bias, such as for example choosing sets governed by experimental setups available at the institute where the experiments are conducted. Utilising numerical optimisation might allow a combination of parameter sets, which might not be intuitively analysed in this typical approach [102].

As discussed in Chap. 8, many different approaches to optimise or improve processes by standard engineering approaches exist in literature. Therefore, to the authors best knowledge, the central aim of this contribution, i.e. the *direct* optimisation of different industrial processes regarding their damage behaviour, has not been thoroughly conducted in literature. Direct minimisation of damage in forming processes is a very complex and computational expensive task. Therefore, damage optimisation in forming is sparsely researched. Specifically, numerical optimisation of extrusion processes has only been rarely considered in literature, mainly due to the specific complexity of the forming process. Nonetheless, numerical damage optimisation for forming processes is considered in some publications.

The most computational taxing approach is to utilise damage modelling to predict the damage behaviour, done for example in [68, 82], which directly or implicitly influence the chosen objective function of the optimisation. Application of numerical optimisation subsequently allows automatic generation of parameter sets which create the desired part with reduced simulated damage. Due to this high computational cost, alternative approaches utilise other measurements to quantify damage and therefore can reduce the modelling cost of the simulation. In [26, 40] for example, the thickness of the sheet material in both processes is chosen as the quantification of damage, or rather failure prevention. Prior experimental investigations, and subsequent modelling of the process, allowed quantifying a threshold variable. Definition of an objective function, allows generating process parameters, which ensure manufacturing of the parts within save margins. A final approach sees the combination of accurate modelling of damage with a meta-models, which are subsequently optimised like in [11, 163]. While this allows optimisation of damage in forming processes, this is not the desired methodology, as was discussed in Sect. 8.3.

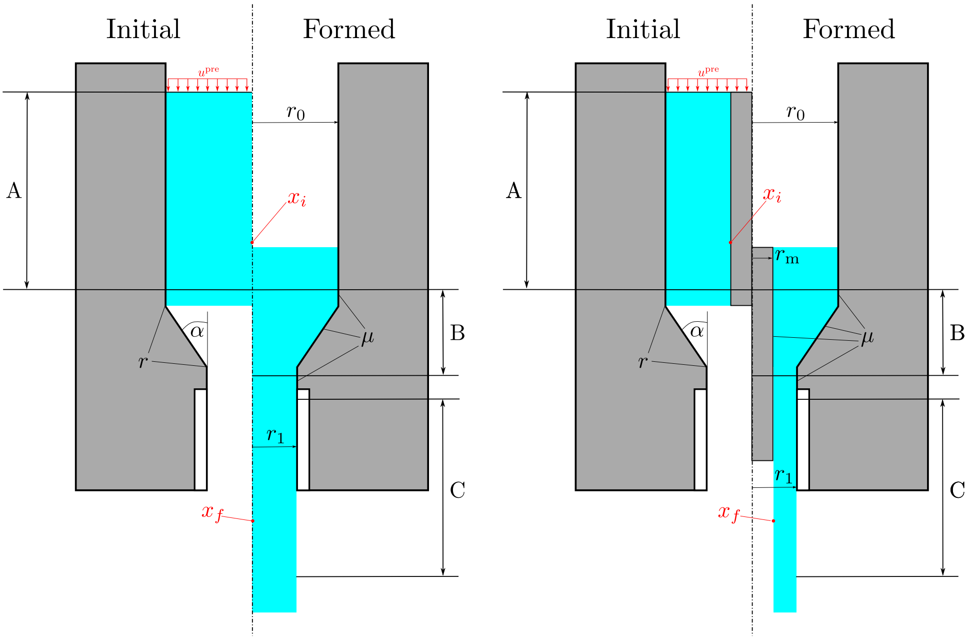
In this chapter, the first two approaches are applied. The stress-triaxiality η, see (8.4), is used to interpret the damage state in a given process, when accurate damage modelling is not utilised. This has proven applicable for the respective problems, see [109, 110, 145, 160]. The damage state is indirectly adjusted by minimising the triaxiality or incorporating it as constraint within the optimisation problem. A reduction in triaxiality value leads to a more compressive dominated stress-state, which is understood to reduce ductile damage accumulation. In addition, a damage model is implemented and the process optimised. The evolution and value of the damage variable can therefore be considered in the optimisation.

Parts of the presented research on numerically optimised forming processes have already been published in [102], for the optimisation of rod extrusion, and in [95, 97, 101] for the different optimisations of bending processes.

10.2 Rod extrusion

In this section the results generated by the optimisation framework, described in the previous chapter Chap. 9, are applied to rod extrusion processes and presented in the following. After an overview regarding the simulation and optimisation model in Sect. 10.2.1, in Sect. 10.2.2 the results from the optimisation framework are compared to research data from literature. This aims to validate the applicability of the framework to bulk forming processes. Following the validation, in Sect. 10.2.3 the framework is used to optimise the process of forward hollow extrusion with different stated optimisation problems in order to generate optimal process parameter sets. Additional data, and a preliminary validation can be found in Appendix D.1. All solutions in this chapter are generated using the gradient-free optimisation within *fminsearch*. The break threshold is the same for all optimisation and the standard value, i.e. $\texttt{tolX} = \texttt{tolFun} = 1 \times 10^{-4}$.

10.2.1 Simulation and optimisation setup



(a) Rod forward extrusion **(b)** Forward hollow extrusion

Figure 10.1: An illustration of the simulations conducted during the optimisation processes. For each image respectively, the left part shows the initial configuration with the formed one on the right.

The Abaqus model for this simulation was provided by Robin Gitschel from IUL, TU Dortmund University and adapted for the application of the optimisation framework as described in Chap. 9. The optimisation is applied to both processes of forward rod extrusion and forward hollow extrusion in the following. Both processes are depicted in Fig. 10.1. The process of forward rod extrusion consists of five relevant parameters regarding the optimisation. These are the initial radius r_0, the final radius r_1, the shoulder angle 2α, the friction parameter μ and the transition radius r, as annotated in Fig. 10.1a). The extrusion strain are computed from both radii, such that $\varepsilon_\mathrm{ex} = \varepsilon_\mathrm{ex}(r_0,r_1)$, see (8.2).

The simulation within Abaqus itself consists of the workpiece and the die. The former is modelled with 20330 linear four-noded thermo-mechanically coupled axisymmetric elements (CAX4RHT). The elasto-plastic material behaviour and its flow curves at 20°C, 200°C and 400°C were obtained in upsetting tests of case hardening steel 16MnCrS5, to model the plastic behaviour within the simulation accurately. The die is initially meshed with 3160 elements (CAX4RT). Since the geometry of the die changes throughout the optimisation, the choice of mostly constant element size was made, such that the number of elements within each optimisation step may vary and is not held constant. While this may lead to small discontinuities in the objective function, no problems were observed while using the gradient-free method. The load is applied as a prescribed deformation at the top part of the workpiece. Contact between the die and the workpiece is of master-slave type, with Coulomb friction, defined by the friction parameter μ between both parts.

Additionally, Fig. 10.1a) depicts the element location where the objective function in Sect. 10.2.2 is evaluated. Its approximate position is denoted with x_i in the initial configuration and x_f after the part is formed. This element has to fulfil certain conditions to allow good comparison with the gathered results in [111]. On the one hand, it has to lie on the central axis of the workpiece, as this area is the most critical regarding damage accumulation. On the other hand, it is chosen such that it lies outside the forming zone at process initiation, transverses the forming area and finally is located within the steady-state region of the forming process after the simulation is finished. These areas of the forming process are labelled in Fig. 10.1a) with the letters A, B and C, respectively.

The process of forward hollow extrusion extends the simulation with the addition of a central mandrel, see Fig. 10.1b). As a result, the set of process parameters for this process additionally includes the mandrel radius r_m, in which case the extrusion strain is a function of the three radii, i.e. $\varepsilon_\mathrm{ex} = \varepsilon_\mathrm{ex}(r_0,r_1,r_\mathrm{m})$, see (8.3). The simulation model itself is generated from the previous process. The mandrel is added as an additional part within the simulation, using the same element type as for the die (CAX4RT). Contact between the mandrel and the workpiece is applied in the same fashion as for the die and the workpiece, i.e. master-slave contact with Coulomb friction. Application of the displacement load at the top of the workpiece also has to be applied to the mandrel. Similarly to the previous process, the number of elements within each part may change throughout the optimisation when altering the design. Consequently, a change in mandrel radius r_m leads to subsequent changes to the inner radius of the workpiece. This in turn leads to a remeshing of the parts, and with the aforementioned constant element size, a change in number of elements. As a result, the problem with choice of sets, stated in Sect. 9.2.3, arises here. Therefore, a geometric set was defined within the workpiece. Within this set, the element for the evaluation of the objective function was derived by sorting through the respective elements, until the element in the lower-left corner of the set was found. This element has to fulfil the previously stated conditions, i.e. it lies outside the forming area initially, traverses it and is finally located in the steady-state region. However, due to the additional mandrel, it no longer can lie on the central axis. Instead, it has to lie at the most central point with respect to the symmetry axis. A graphical illustration for this location is depicted in Fig. 10.1b). The initial location is denoted with x_i and the final formed location with x_f. All important process parameters with respect to the following optimisations are summarised in Table 10.1.

10.2.2 Optimisation of forward rod extrusion

In order to validate the optimisation framework, the resulting set of optimal process parameters are compared to the existing research. To this extent, the first optimisation problem is to optimise the forming process of forward rod extrusion. The design variables

Table 10.1: Names and variables of the process parameters for their respective optimisations.

Forward rod extrusion		
symbol	variable name	design variable
$\varepsilon_{\mathrm{ex}}$	extrusion strain	yes
2α	shoulder angle	yes
r_0	initial radius	no
r_1	final radius	no
r	transition radius	yes
μ	friction parameter	yes
Forward hollow extrusion		
symbol	variable name	design variable
$\varepsilon_{\mathrm{ex}}$	extrusion strain	yes
2α	shoulder angle	yes
r_0	initial radius	no
r_1	final radius	yes
r_{m}	mandrel radius	yes
r	transition radius	no
μ	friction parameter	no

$\mathbf{s}$ for this validation encompass the complete set of process parameters of the simulated process as stated in Sect. 10.2.1, i.e. $\mathbf{s} = \{\varepsilon_{\mathrm{ex}}, 2\alpha, r, \mu\}$. The optimisation problem solved for this purpose thus reads

$$\underset{\mathbf{s}_{\mathrm{l}} \leq \mathbf{s} \leq \mathbf{s}_{\mathrm{u}}}{\text{minimise}} \quad \mathfrak{J}^{\mathrm{re}} = \eta^{\mathrm{mean}}(\mathbf{s}). \tag{10.1}$$

The objective of this problem is to minimise the mean stress-triaxiality η^{mean}, see (8.5), for a specified element over the course of the complete simulation, wherein a lower mean stress triaxiality coincides with lower damage accumulation, see Chap. 8. Since the extrusion strain $\varepsilon_{\mathrm{ex}}$ is a function of the initial radius r_0 and final radius r_1, with the former remaining constant, the change within the framework is achieved by changing the final radius of the forming process. For better comparison however, the extrusion strain is chosen as a design variable for the optimisation and subsequent comparison. A general overview of the optimisation problem and its results are summarised in Table 10.2. Therein, the design

Table 10.2: Overview of the optimisation of full forward rod extrusion.

$\mathbf{s}$	$\mathbf{s}_{\mathrm{l}}$	$\mathbf{s}_{\mathrm{u}}$	impact ($\Delta\eta^{\mathrm{mean}}$)	initial (k=0)	intermediate (k=8)	optimum (k=93)
$\varepsilon_{\mathrm{ex}}$	0.3	1.5	high (↓)	0.7	1.279	1.496
$2\alpha/°$	30	150	medium (↑)	60	30.226	35.857
r/mm	2	5	low (↓)	3	3.301	2.036
μ	0.04	0.06	very low (↓)	0.046	0.048	0.059

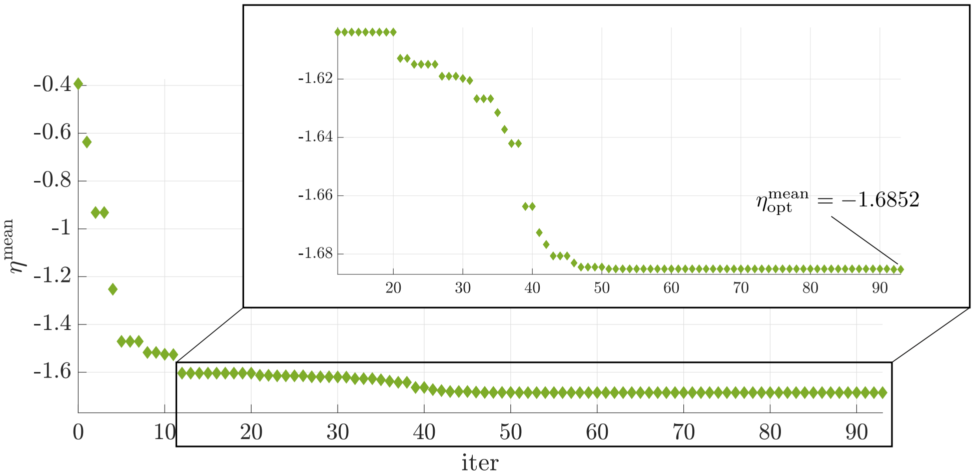


Figure 10.2: The iteration process over the objective η^{mean} for the validation of the framework by optimising the process of full forward rod extrusion. The boxed in area shows a zoomed-in depiction for the respective iterations with a rescaled y-axis.

variables $\mathbf{s}$ and the lower and upper bounds, $\mathbf{s}_{\mathrm{l}}$ and $\mathbf{s}_{\mathrm{u}}$, are listed. These values are chosen to be in concordance with the analysed experiments in [111]. The fourth column shortly summarises the results of the cited paper onto the analysed change in mean triaxiality η^{mean}. Increasing the extrusion strain for example leads to a decreased mean triaxiality value, denoted by the down-arrow. In contrast, an increase in the shoulder angle increases the mean triaxiality, denoted by the up-arrow. Furthermore, the extrusion strain has a "high" impact on the evaluated mean triaxiality, compared to e.g. the "medium" impact of the shoulder angle, and thus leads to a bigger change in the objective value. The fifth column lists the chosen initial values for the design variables. The optimal results are shown in the last column. The optimised parameters show a good accordance with the research data. The extrusion strain $\varepsilon_{\mathrm{ex}}$ and the friction parameter μ are maximised and the shoulder angle 2α minimised, which lead to the lower mean triaxiality $\eta_{\mathrm{opt}}^{\mathrm{mean}} = -1.6852$, compared to the initial value of $\eta_{\mathrm{init}}^{\mathrm{mean}} = -0.393$. Even though three parameters show good accordance, the transition radius r does not. While throughout the optimisation the parameter increases, as depicted in the sixth column, the optimiser reduces it when converging to the optimal solution. Comparing this to the cited research, it should instead increase in value. However, since the general "low" impact this parameter has on the objective value overall, the validation of the optimisation framework is deemed successful. Most importantly, the optimiser initially changes the extrusion strain $\varepsilon_{\mathrm{ex}}$ and the shoulder angle 2α, as they are generally of most importance for this process when dealing with minimised ductile damage accumulation in the formed workpiece. This is represented in column six, depicting early iteration values of those parameters during the optimisation procedure.

Additionally, the iteration process is depicted in Fig. 10.2. It shows the rapid change in objective function due to the previously mentioned initial change of extrusion strain and shoulder angle. However, since a gradient-free method is used to optimise the process, the optimisation takes a long time to converge even for such a small number of design variables. This is mainly due to the chosen relative tolerances for the parameters and the objective function, see Sect. 10.2.

10.2.3 Optimisation of forward hollow extrusion

Following the successful validation of the framework, the optimisation is now expanded to the more complex forming process of forward hollow extrusion. This process includes a mandrel to produce circular hollowed workpieces. Consequently, the extrusion strain is now a function of the initial radius r_0, the final radius r_1 and additionally the newly introduced mandrel radius r_{m}. However, to ease computation of this more complex forming process, the transition radius $r = 1\,\mathrm{mm}$ and the friction parameter $\mu = 0.04$ are held constant throughout and not included as design variables the optimisation, see again Table 10.1.

Optimisation with a constant mandrel radius

For the first optimisation problem, the extrusion strain $\varepsilon_{\mathrm{ex}}$ and the shoulder angle 2α are chosen as design variables. To allow a comparison to the previous process, the mandrel radius is initially held constant, i.e. $r_{\mathrm{m}} = 6.7\,\mathrm{mm}$. Therefore, the design variables for this problem are $\mathbf{s}_1 = \{\varepsilon_{\mathrm{ex}}(r_1),2\alpha\}$, leading to the following optimisation problem

$$\underset{\mathbf{s}_{\mathrm{l}} \leq \mathbf{s}_1 \leq \mathbf{s}_{\mathrm{u}}}{\text{minimise}} \quad \mathfrak{J}^{\mathrm{re}} = \eta^{\mathrm{mean}}(\mathbf{s}_1). \tag{10.2}$$

This first optimisation problem aims to determine whether comparable statements can be made about the selected process parameters between the hollow and forward rod extrusion process, i.e. the extrusion strain $\varepsilon_{\mathrm{ex}}$ and the shoulder angle 2α. The objective function remains the same as in the validation of forward rod extrusion, i.e. the mean stress triaxiality η^{mean}. The optimal result for this process seems to indicate similar behaviour, see Table 10.3. The extrusion strain gets maximised and the shoulder angle minimised, leading to swift decrease in objective value. Due to the added mandrel, even for a lower maximum extrusion strain of $\varepsilon_{\mathrm{ex}} = 1.4$, the objective function yields a lower value than in the forward rod extrusion process, indicating a better behaviour regarding damage evolution. Looking at the iteration graph in Fig. 10.3, the optimiser swiftly approaches a value of $\eta^{\mathrm{mean}} = -1.7667$, which is close to the optimal solution. However, due to the choice of solver and the break condition, further 40 iterations are necessary to achieve convergence, where the change of the process parameters is rather small within those final steps, see Table 10.3.

Optimisation with a variable mandrel radius

The second optimisation problem now includes a variable mandrel radius r_{m}. For this reason, the extrusion strain $\varepsilon_{\mathrm{ex}}$ is no longer utilised as an explicit design variable. Instead, the final radius r_1 is now included. The extrusion strain $\varepsilon_{\mathrm{ex}}$ therefore now enters as an additional constraint. The design variables for the second optimisation problem thus

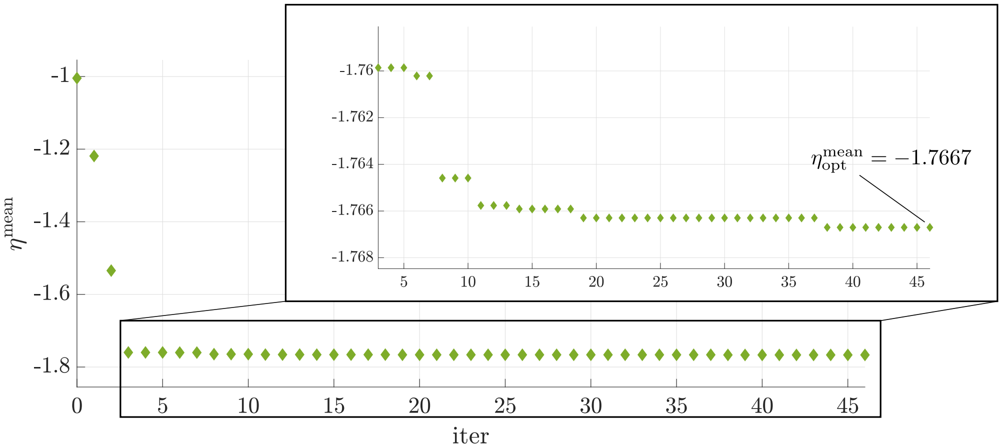


Figure 10.3: Objective function over iterations for the first optimisation problem of forward hollow extrusion with a constant mandrel radius r_{m}. The boxed in area shows a zoomed-in depiction for the respective iterations with a rescaled y-axis.

Table 10.3: The first optimisation problem of forward hollow extrusion with a fixed mandrel radius r_{m}.

$\mathbf{s}$	dependent	fixed	$\mathbf{s}_{\text{l}}$	$\mathbf{s}_{\text{u}}$	initial (k=0)	intermediate (k=8)	optimum (k=46)
-	-	r_0/mm	15.1	15.1	15.1	15.1	15.1
-	r_1/mm	-	9.484	13.957	13.957	9.491	9.484
-	-	r_{m}/mm	6.7	6.7	6.7	6.7	6.7
ε_{ex}	-	-	0.2	1.4	1	1.397	1.399
$2\alpha/°$	-	-	30°	150°	90	36.540	36.121

encompass $\mathbf{s}_2 = \{r_{\text{m}}, r_1, 2\alpha\}$, with the subsequent optimisation problem

$$\begin{aligned} &\underset{\mathbf{s}_{\text{l}} \leq \mathbf{s}_2 \leq \mathbf{s}_{\text{u}}}{\text{minimise}} && \mathfrak{J}^{\text{re}} = \eta^{\text{mean}}(\mathbf{s}_2) \\ &\text{subject to} && 0.2 \leq \varepsilon_{\text{ex}}(r_{\text{m}}, r_1) \leq 1.4. \end{aligned} \tag{10.3}$$

The resulting, and some intermediate values, are listed in Table 10.4. As expected, the optimiser tries to maximise the extrusion strain initially, since this has the highest impact on the objective function. This was already derived in the previous optimisation problems. Since this value now depends on two possible changes in the chosen design variables, intermediate results show that at first the final diameter r_1 is decreased, with the mandrel radius r_{m} also being increased to achieve a high extrusion strain. The shoulder angle remains mostly constant at the beginning and only decreases at later stages in the optimisation, indicating a lower impact on the objective value compared to the radii. After increasing the extrusion rate, the optimiser again increases the final radius r_1, while further increasing the mandrel radius r_{m} in order to remaining at the bounded extrusion

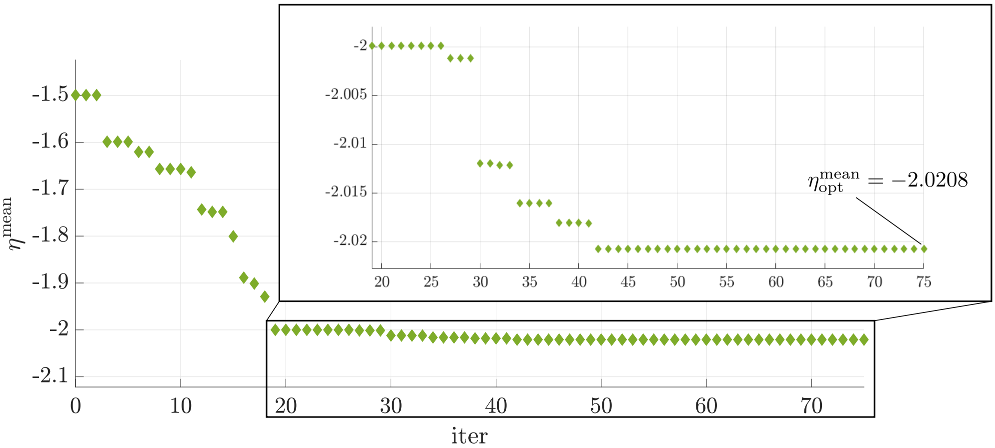


Figure 10.4: Objective function over iterations for the second optimisation problem of forward hollow extrusion with a variable mandrel radius r_{m}. The boxed in area shows a zoomed-in depiction for the respective iterations with a rescaled y-axis.

rate value of $\varepsilon_{\text{ex,u}} = 1.4$. This finally leads to the optimal design stated in Table 10.4. The iterative process in Fig. 10.4 also shows the described behaviour. Throughout the optimisation, up to iteration 19, large jumps in the objective function can be seen, mostly due to the changes in radii. At this point, the design variables are close to the final optimal values and only small adjustments are made, leading to consequently small modification of the objective function, depicted in the zoomed-in part of Fig. 10.4.

Table 10.4: The table for the second optimisation of forward hollow extrusion with a variable mandrel radius r_{m}.

s	dependent	fixed	s_{l}	s_{u}	initial (k=0)	intermediate (k=12)	optimum (k=75)
-	-	r_0/mm	15.1	15.1	15.1	15.1	15.1
r_1/mm	-	-	9.5	13.9	11.5	11.316	11.815
r_{m}/mm	-	-	6.7	11.6	9.5	9.703	10.517
-	ε_{ex}	-	0.2	1.4	1.188	1.373	1.399
2α/°	-	-	30°	150°	60	58.839	30.012

Of note here is, that the final values of r_1 and r_{m} are not at either respective boundary, indicating that it may be beneficial for this forming process to choose tool sets generated by the optimisation. This behaviour is highlighted in Fig. 10.5, which illustrates the iterative behaviour of the mandrel radius r_{m} and the final radius r_1 over the iterations of the optimisation. Additionally, the respective resulting extrusion strain ε_{ex} is presented as well. Finally, the upper bounds for the respective radii are indicated to highlight that these bounds never explicitly apply to their respective design variables throughout the optimisation. The figure only shows the behaviour up to iteration 42. After that point in the optimisation, no notable changes in design occur, which is also detectable in the little

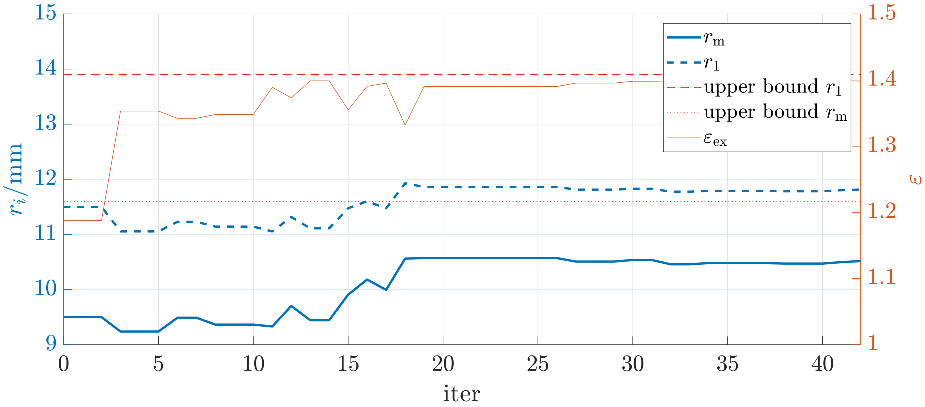


Figure 10.5: An iterative display of the mandrel radius r_m and the final radius r_1 over the iterations up to iteration 42. Additionally, their respective upper bounds and the resulting extrusion strain ε_ex for each iteration are depicted.

development of the objective function, again see Fig. 10.4. After a few initial iterations, the optimiser starts to increase the extrusion strain by, at first, reducing the respective radii. The depicted initial constant design variables, i.e. the constant values of r_m and r_1 for iterations 0 to 2, are a result of the Nelder-Mead simplex algorithm. As for this implementation, only the current optimal solution of each iterative step is stored. This might be in turn the initial starting guess as other points of the simplex may yield higher objective values. The following change to the extrusion strain of $\varepsilon_\mathrm{ex} = 1.35$ at iteration 3 is achieved by reducing the mandrel and final radius, respectively. After this initial change, the radii show some positive and negative adjustments up to iteration 12. At this point in the optimisation, the behaviour regarding the radii changes drastically and their respective values start to increase in value by up to 1 mm. The peak of this behaviour is reached at iteration 18. From there on the optimiser only makes small changes and again decreases their respective values. This coincides with the small changes of the objective function, as was discussed regarding Fig. 10.4. After iteration 26, the radii get reduced again which in turn increases the extrusion strain. This behaviour repeats until iteration 41, after which a final small increase in design is visible. A few intermediate iterations show points, where the extrusion strain gets reduced. Most notably this occurs at iterations 12, 15 and 18, respectively. At these iterations the iterative optimal solution showed a lower shoulder angle, which in turn benefits the triaxiality value. Subsequently, this angle remains constant for a few iterations while the extrusion strain increases again. A value close to the optimal, i.e. $2\alpha = 30.162°$, is finally reached at iteration 18. Afterwards, the extrusion strain reaches a value of over $\varepsilon_\mathrm{ex} = 1.39$ again and does not fall below that threshold throughout the rest of the optimisation.

Additional calculations were performed to underline this result, i.e. the objective having a lower value between the bounds of the radii rather than at either limit. Further optimisations with different starting values at the upper boundaries however deemed difficult since the optimiser often ran into iterative solutions which did not converge for the finite element (FE) problem due to high mesh distortions. However, at best, the possible manual calculations, as well as intermediate results from the additional optimisations, yielded results in the area of $\eta^{\text{mean}} = -1.999$, indicating an overall better solution generated by the optimisation as described above.

10.3 Bending

In this section, the enhanced bending process of elastomer bending is optimised, cf. [221]. As introduced in Sect. 8.2, the critical area regarding damage accumulation is the outer bending fibre of the sheet metal. The inclusion of the elastomer allows to induce compressive residual stresses in the outer fibre of the sheet and thereby delay the onset of damage evolution, cf. [160, 161, 221]. Further optimisation results for bending, i.e. load and load path optimisation, can be found in Appendix D.2. The described process, and the other approaches followed in the aforementioned appendix, are depicted in Fig. 10.6.

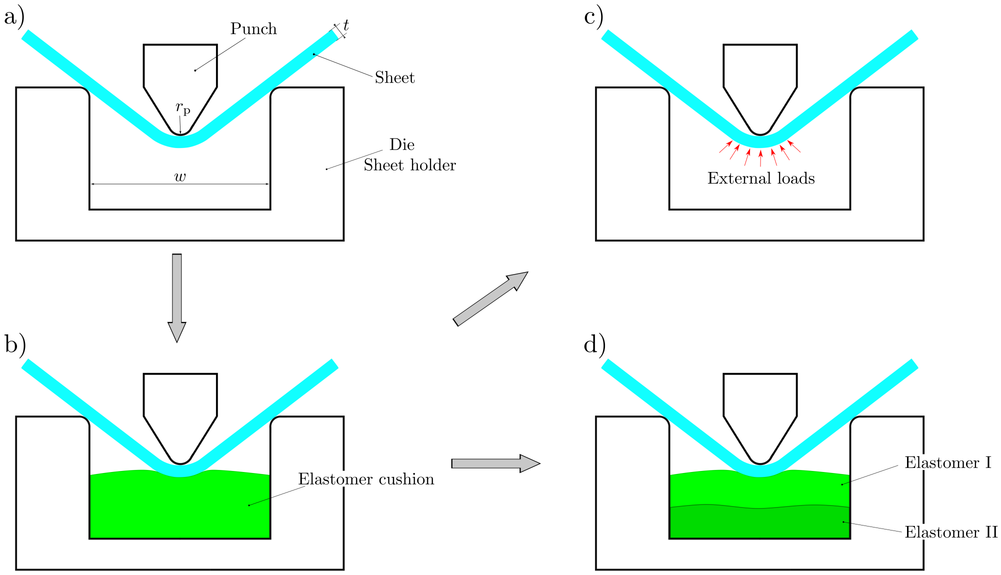


Figure 10.6: The left side shows a) free air bending in its typical configuration and the advanced b) elastomer bending. Based on these processes, the two right images depict the approaches for the optimisation in Appendix D.2: c) The load optimisation with external loads and d) the optimisation of the elastomer cushion with different elastomer types.

10.3.1 Simulation model

The simulation for the optimisation of elastomer bending relies on the same model, as the one used in the benchmark section of the previous chapter, see Sect. 9.3.1. To summarise, the sheet has a thickness of $t = 1.2\,\text{mm}$, the die a width of $w = 24\,\text{mm}$ and the die and punch a radius of $r_{\text{p}} = r_{\text{d}} = 1\,\text{mm}$. The process is performed with a prescribed punch displacement of $u_{\mathrm{punch}} = 5\,\mathrm{mm}$.

New in regard to this section and elastomer bending, is the inclusion of an elastomer cushion. This is placed inside the die, see Fig. 10.6b), and is modelled as a deformable body. A hyperelastic material model of Mooney-Rivlin type is assumed for the elastomer, cf. [58], with the parameters for the two types of elastomer used in the following optimisation listed in Table 10.5. Therefore, additional contact interactions are included into the simulation, i.e. elastic-rigid contact between elastomer and die, as well as elastic-elastic contact between sheet and die. The latter is crucial for the optimisation, as it includes many more discontinuities into the problem and therefore the mathematical description. The elastomer is meshed with 3600 elements of type CPS4R. The simulation model has been provided by the original author from IUL, TU Dortmund University.

Table 10.5: Material parameters for the two types of elastomer.

elastomer type	C_{10}	C_{01}	D_1
Type I (hard)	1.92521389	0.967182642	0
Type II (soft)	1.67336094	14.6497159	0

10.3.2 Elastomer optimisation

In the following, the process of elastomer bending from [221] is directly optimised, by adjusting the composition of the inserted elastomer layers. For this purpose, the elastomer cushion within the simulation is split into two separate sections of one hard elastomer type and one soft elastomer type, i.e. of 75% hard and of 25% soft type, see Fig. 10.7. The objective of the optimisation is to generate an homogeneous triaxiality distribution in the outer fibre of the sheet where it is in contact with the elastomer. In the initial case, the triaxiality gets reduced the most in the middle of the process where the punch pushes down onto the sheet, i.e. at $x = 0$ for the local coordinate system in Fig. 10.7. Since the highest vertical displacement of the sheet occurs at that location, this results in the highest displacement for the elastomer as well. Consequently, the reaction forces due to the elastomer maximise in this area, and therefore the triaxiality is reduced the most.

A more even distribution of triaxiality allows for lower localisation of damage and therefore reduces the risk of failure while forming. This is captured by a least-square problem between the triaxiality values in the selected area, and a prescribed critical triaxiality value, wherein again $\eta_{\mathrm{crit}} = 0.48$ is chosen. Thus, the optimisation problem reads

$$\underset{\mathbf{s}_{\mathrm{l}} \leq \mathbf{s} \leq \mathbf{s}_{\mathrm{u}}}{\mathrm{minimise}} \quad \mathfrak{J}^{\mathrm{Geom}} = \|\eta(\mathbf{s}) - \eta_{\mathrm{crit}}\|^2. \tag{10.4}$$

The choice of optimised area within the sheet has to be considered carefully. Since the reduction of triaxiality due to the elastomer is not evenly distributed over the width and thickness, the choice of the homogenised area heavily influences the results, similarly to the discussion of the benchmark springback optimisation in Sect. 9.3.3. Defining the width and/or thickness of this area too large, will result in unsatisfactory homogenisation, as large fluctuations in triaxiality will occur. If the chosen thickness is too high, the pressure resulting from the optimised elastomer will be higher, resulting in even lower triaxiality values in the outer fibre of the sheet. Therefore, the desired homogenisation is not achievable. The same problem occurs when setting the width of the optimisation area too high. Since the effect of the elastomer gets reduced with higher offset from the centre of the sheet, large changes in elastomer geometry are necessary to inflict even slight changes in elastomer. This again has the inverse effect in the centre of the sheet and therefore the same outcome of bad homogenisation will arise.

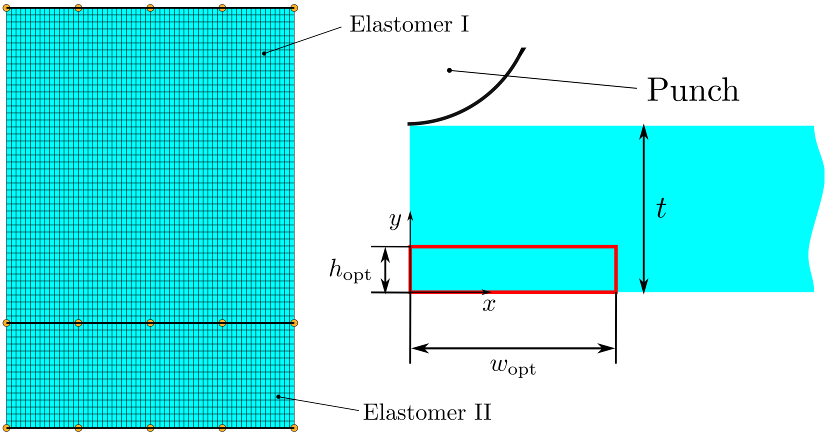


Figure 10.7: The initial CAGD with the 75% and 25% split on the left. The right depicts the chosen area for triaxiality homogenisation within the sheet.

The chosen area for the optimisation has to be pretty small in order for the optimiser to yield satisfying results. Consequently, it only has a width of $w_{\mathrm{opt}} = 0.65\,\mathrm{mm}$ and height $h_{\mathrm{opt}} = 0.15\,\mathrm{mm}$, see Fig. 10.7. The small chosen height is especially necessary, as choosing to many elements, in this case 2 in sheet thickness direction, will make it nearly impossible for the optimiser to converge. The triaxiality has a significantly higher gradient in thickness direction and therefore setting the objective area to large will significantly adjust the results. In addition, the initial Computer Aided Geometric Design (CAGD) is presented showing the split in elastomer distribution. The initial guess is chosen to be of 75% hard elastomer and 25% soft elastomer. Only the five knots which divide the two elastomer parts are chosen as design variables; the remaining nodes remain fixed.

The simulation model was provided by the authors of [221] from IUL, TU Dortmund University. With the addition of the elastomer, the contact interactions are extended with contact between sheet and elastomer, as well as die and elastomer. The friction coefficient of Coulomb type friction for these interactions is set to $\mu = 0.5$. To include the geometry optimisation into the setup, the CAGD was created by a Bézier-surface, using the same code as in Chap. 6. To transfer data between the Matlab and Abaqus environment, since a new mesh is generated in Matlab, the model of the elastomer within Abaqus as well as in Matlab has to coincide. This enables a mapping of the coordinates between the two simulation software, such that

$$\mathbf{X}_{\mathrm{Abq}} = \mathbf{M} \cdot \mathbf{X}_{\mathrm{Mat}}, \tag{10.5}$$

where $\mathbf{X}_\bullet$ are the nodal coordinates within each software, and $\mathbf{M}$ a mapping matrix that has to be deduced in the beginning of the optimisation. After each iteration, the new mesh generated by the CAGD is mapped back to Abaqus, allowing the iterative design to be incorporated into the optimisation. For sake of simplicity, and to improve the optimisation behaviour as well as simulation setup, the two types of elastomers are connected as one body within the simulation. This body is split into two sections with the different material properties of the respective elastomer type. In reality, the two types of elastomers would have to be placed on top of each other or glued together. However, simulating this with contact mechanics for example would increase the computational complexity and might lead to further convergence issues within the optimisation.

Initially, the optimisation was run with SQP, similarly to the other bending optimisation problems in this work, see Sect. 9.3 and Appendix D.2. However, the algorithm was not able to converge to a minimum. Instead, the algorithm was alternating between two solutions. The reason behind this behaviour is most likely due to the numerically derived gradients. With the inclusion of the elastomer cushion, and the contact interactions between elastomer and die, and more importantly the elastic-elastic interaction between elastomer and sheet, the numerically computed gradients yield too high inaccuracies for the optimisation procedure. To solve this problem, again the Nelder-Mead simplex is applied. This enables derivation of a unique solution which is depicted in Fig. 10.8 together with the initial guess. The results presented show the possible homogenisation of the triaxiality, and further enhancement of the process. In the beginning, due to the inclusion of softer elastomer into the process, the triaxiality in the sheet exceeds the critical triaxiality $\eta_{\text{crit}} = 0.48$. The optimal elastomer design decreases the maximal triaxiality values and further homogenises the surrounding area. To achieve this, the initial flat transition area between hard and soft elastomer gets altered. A wave-like curvature is created, which is thicker to the left, decreases between the second and third design variable, again creases at the fourth design variable and then decreases in thickness again. Additionally, the overall amount of soft elastomer gets reduced and therefore more of the harder elastomer is part of the cushion.

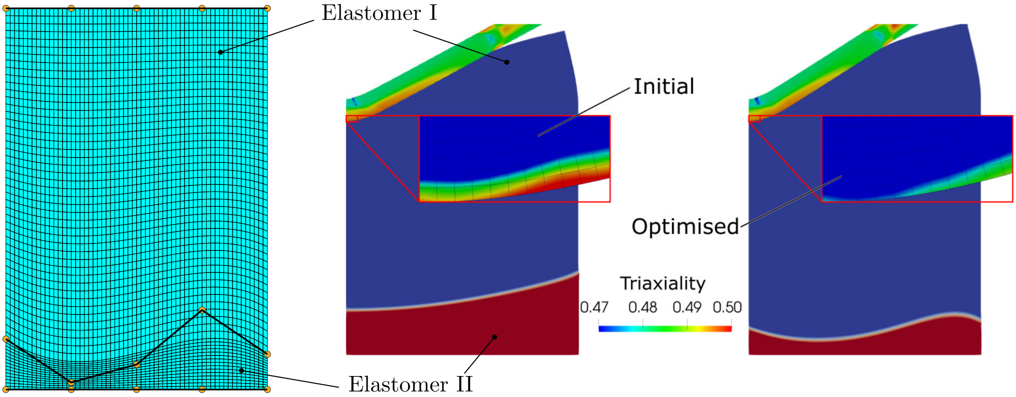


Figure 10.8: Depicted are the results of the elastomer optimisation. The left shows the CAGD of the elastomer, the right the simulation results. The elastomer types are depicted in red (soft) and blue (hard). The zoomed in area shows the homogenised area of the sheet metal.

While the presented results are promising, some shortcomings are detectable. The homogenisation with this optimisation framework is improved, but not yet perfectly even over the selected area. Additionally, the triaxiality in the centre of the sheet, i.e. at the symmetry y-axis, is still significantly lower than further to the right. Nonetheless, the presented results highlight the possibilities of this framework regarding an optimisation of this process. Enhancements, such as finer discretisation of design variables in the CAGD, or additional layers of different elastomer types, could enhance the optimisation problem and allow an even more homogenised stress-state in the sheet.

Furthermore, in Sect. 9.2.3 it was mentioned that this approach of mapping meshes between external mesh $\mathbf{X}_{\text{mat}}$ and the one in Abaqus $\mathbf{X}_{\text{Abq}}$ in (10.5) can lead to difficulties. This is detectable in the presented results. Initially, the elements all have the same size, see Fig. 10.7. However, due to this mapping, the optimal shapes generate a very distorted mesh in the lower section for Elastomer II, see Fig. 10.8. This further emphasises the benefit of the *MDB* approach described in Sect. 9.2.3, as it was later adopted for IOPS.

10.4 Stretch indenting

In this section the last forming process is considered for damage optimisation. Instead of optimising deep-drawing, the process of stretch indenting is chosen instead. It is very similar to classical deep-drawing and allows 2d modelling, assuming rotational symmetry. The material flow within in the process is adjusted by a drawbead, which is optimised in the following. The drawbead restricts the material flow, leading to a thinning at the bottom of the drawn cup. Since deep-drawing processes are very quick processes, the resulting time-dependency has to be taken into consideration within the simulation environment and the process can no longer be assumed to be quasi-static.

10.4.1 Simulation model

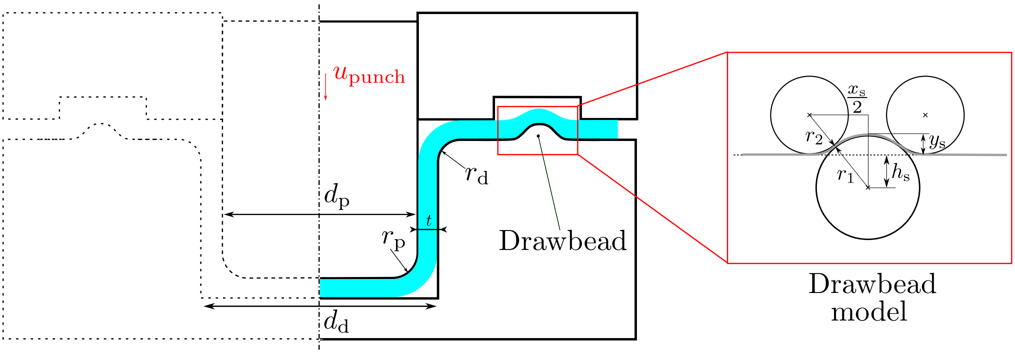


Figure 10.9: The forming process stretch indenting. The dashed part of the process is not simulated. Additionally, the drawbead model and the necessary parameters for its generation within the simulation.

The simulation model for the process is sketched in Fig. 10.9, wherein the dotted part of the process is omitted within the simulation by exploiting the symmetry conditions and has been kindly provided by WZL, RWTH Aachen University. The geometries of the process are as follows: The punch has a diameter of $d_{\mathrm{p}} = 25\,\mathrm{mm}$, the die a diameter of $d_{\mathrm{d}} = 26.8\,\mathrm{mm}$ and the transition radii of punch and die have a radius of $r_{\mathrm{p}} = r_{\mathrm{d}} = 3\,\mathrm{mm}$. Research shows, that these radii have an impact on the damage behaviour, cf. [165, 166], but since the focus is placed onto the drawbead, will be omitted as possible design variables here and held constant throughout the optimisation. The sheet plate has a thickness of $t = 1.5\,\mathrm{mm}$. The FE model is run with Abaqus/Explicit, an explicit solver which enables the calculation of dynamic problems. Since deep-drawing processes cannot by run accurately for quasi-static cases, as these problems happen in a fraction of a second, a dynamic solver and setup are necessary.

The simulation model consists of two steps. In the first step, the blank holder is pressed onto the sheet and therefore restrict vertical movement and initiates a first deformation of the sheet due to the geometry of the drawbead. The second simulation step is the actual forming process, wherein the punch presses down onto the sheet metal. A prescribed velocity of $v_{\mathrm{punch}} = 50\,\frac{\mathrm{mm}}{\mathrm{s}}$ is applied to the punch over $t_{\mathrm{sim}} = 0.244\,\mathrm{s}$, resulting in a displacement of $u_{\mathrm{punch}} = 12.2\,\mathrm{mm}$. However, the simulation is setup such that initially the sheet is offset by 6.2 mm with respect to the die. Taking this into account, the sheet only gets drawn for 6 mm. Due to the drawbead, the material flow is adjusted, resulting in different damage behaviour.

The material model used is a local Lemaitre damage model, which has been calibrated for use in modelling of DP800 in [216]. Due to the nature of the mesh-dependence in local damage models, the mesh of the sheet has to coincide with the mesh of the calibration method. As such, the elements for the sheet have an edge-size of 0.5 mm and the complete sheet is meshed with 408 elements of type CAX4R. Similar to the previous processes, master-slave type contact is assumed between die and sheet, punch and sheet, as well as blank holder and sheet. The friction is assumed to be Coulomb type with the coefficient at these contact surfaces set to $\mu = 0.05$, as these kinds of processes are lubricated and therefore the parameter is set small.

The most important part of the simulation regarding optimisation is the model of the drawbead. It is defined with three circles, see Fig. 10.9. A continuos line is drawn between these circles, which forms geometrical shape of the drawbead in the die. The upper two radii have the value r_2 and the lower one the radius r_1. The drawbead has a width of x_{s} and a height of y_{s}. Additionally, an offset h_{s} for the lower radius r_1 is included, which increases the design space for the drawbead geometry.

10.4.2 Drawbead optimisation

The objective of the optimisation is to reduce the damage accumulation within the process after forming, i.e. the prescribed time of $t_{\mathrm{sim}} = 0.244\,\mathrm{s}$. Two optimisation problems are proposed with different objective functions to describe the desired problem of damage minimisation. The first objective only takes into account the maximal value of nodal damage data, i.e. $\mathfrak{J}^{\max} = d_{\max} = \max(\mathbf{d}(\mathbf{s}))$. While this objective leads to a discontinuous optimisation problem, the following optimisations are solved using the gradient-free Nelder-Mead simplex. Similarly to previous results in rod extrusion and elastomer bending, optimisation with gradient-based methods is not suitable for this process. Early tests resulted in non-converging optimisations or optimal solutions being equal to the starting guess.

The second optimisation problem uses the damage accumulation captured by the norm of the nodal damage values. This approach was described in Sect. 5.6 of Part I of this thesis. This description allows capturing the vector of nodal damage values in a scalar valued objective function. The second objective therefore reads $\mathfrak{J}^{\mathrm{norm}} = \|\mathbf{d}(\mathbf{s})\|^2$. While the local damage stores the damage quantities at Gauss-point level, Abaqus allows extrapolation of this data to the nodes and is therefore accessible for the formulation of this objective.

Both optimisation problems are subjected to a set of manufacturing constraints. The general criteria, based on [33], are that the space between the drawbead and the point where the die radius r_{d} initiates, as well as the drawbead and the outer edge of the sheet metal, must be at least 5 mm wide. Applying these conditions, together with the centre point of the position of the drawbead within the die and the total size of the die, constrains the available design space for the drawbead width x_{s}. This ultimately leads to maximum width of the drawbead of $x_{\mathrm{s}} \leq 22.2\,\mathrm{mm}$ for the optimisation. In addition, certain restrictions are placed onto the radii which define the drawbead. The drawbead must be taller than the sheet metal, which leads to a lower bound of 2 mm for the radii and drawbead height. Additionally the two radii must not exceed the radius of die $r_{\mathrm{d}} = 9\,\mathrm{mm}$. Another constraint is the drawbead height must not exceed four times the sheet thickness. Together, this leads to the bounds and constraints

$$2 \leq r_1 \leq 9 \tag{10.6}$$
$$2 \leq r_2 \leq 9 \tag{10.7}$$
$$2 \leq y_{\mathrm{s}} \leq 6 \tag{10.8}$$

Using this information, another constraint is placed upon the drawbead offset h_{s}. First, the offset must not be negative as this would put the centre of r_1 outside the die and lead to an undercut of the drawbead. Additionally, the upper limit of the offset is equal to radius r_1 minus the lower bound of the drawbead height, i.e. 2 mm. This leads to the additional constraint

$$0 \leq h_{\mathrm{s}} \leq r_1 - 2. \tag{10.9}$$

The two optimisation problems are thus defined as

$$\begin{aligned} \underset{\mathbf{s}_{\mathrm{l}} \leq \mathbf{s} \leq \mathbf{s}_{\mathrm{u}}}{\text{minimise}} \quad & \mathfrak{J}^{\max} = d_{\max}(\mathbf{s}) \\ \text{subject to} \quad & x_{\mathrm{s}} \leq 22.201 \\ & y_{\mathrm{s}} \leq 6 \\ & h_{\mathrm{s}} \leq r_1 - 2, \end{aligned} \tag{10.10}$$

and

$$\begin{aligned} \underset{\mathbf{s}_\mathrm{l}\leq\mathbf{s}\leq\mathbf{s}_\mathrm{u}}{\text{minimise}} \quad & \mathfrak{J}^{\mathrm{norm}} = \|\mathbf{d}(\mathbf{s})\|^2 \\ \text{subject to} \quad & x_\mathrm{s} \leq 22.201 \\ & y_\mathrm{s} \leq 6 \\ & h_\mathrm{s} \leq r_1 - 2. \end{aligned} \tag{10.11}$$

Notably, both optimisation problems do not include the lower constraint of the drawbead height $2 \leq y_\mathrm{s}$. This is implicitly included in the lower constraint $h_\mathrm{s} \leq r_1 - 2$, since

$$h_\mathrm{s} \leq r_1 - 2 \Leftrightarrow 2 \leq r_1 - h_\mathrm{s} \Leftrightarrow 2 \leq y_\mathrm{s}, \tag{10.12}$$

by applying the trigonometric properties of the drawbead model, see Fig. 10.9. Additionally, these properties allow the computation of the drawbead height and width with the design variables $\mathbf{s} = \{r_1, r_2, h_\mathrm{s}\}$, such that

$$x_\mathrm{s} = 2\sqrt{[r_1 + r_2]^2 - [r_2 + h_\mathrm{s}]^2}, \tag{10.13}$$

$$y_\mathrm{s} = r_1 - h_\mathrm{s}. \tag{10.14}$$

Even though the drawbead offset h_s is a design variable of the process, an upper constraint is applied. Since this constraint is dependent on another design variable, it cannot be defined in boxed constraints and has to be additionally checked within each iteration together with the other constraints.

Initially seven initial guess were calculated for each optimisation problem (10.10) and (10.11). The first four were randomly defined sets of design parameters, while the latter three describe three different shapes of the drawbead, i.e. a small one, a large one and an average one. The results for these optimisation problems are depicted in their iterative graphs in Fig. 10.10. Due to the amount of data, the explicit results from the computations are found in Appendix D.3. Analysing the data, there is no conclusive data regard the design variables themselves, i.e. r_1, r_2 and h_s, neither does the drawbead width x_s noticeably influence the damage behaviour of the process. However, notably the drawbead height y_s allows a concise conclusion. In all the computed optimisations, excluding run x06 for problem (10.10) which most likely ran into a local minimum, the drawbead height gets reduced close to its minimal value of $y_\mathrm{s}^{\mathrm{min}} = 2\,\mathrm{mm}$. This holds true regardless of the choice of $\mathfrak{J}^{\mathrm{max}}$ or the normed value $\mathfrak{J}^{norm}$. With this knowledge, the significant improvement in damage reduction for specimen x04 is obvious. Initially, the drawbead height for this initial guess is extremely large at $y_\mathrm{s} = 6\,\mathrm{mm}$, see Fig. 10.11. Due to this, the value of maximal damage can be reduced significantly by 21% from $d_{\mathrm{max}}^{\mathrm{init}} = 0.32883$ to $d_{\mathrm{max}}^{\mathrm{opt}} = 0.26003$.

Based on this research, a final set of four optimisations were conducted, which used starting parameters close to the optimal value for $y_\mathrm{s} = 2\,\mathrm{mm}$, aiming to find better a combination of the other process parameters. However, while a better value for d_{max} could be found, the effect was negligible small. Therefore, the general conclusion for this process regarding damage tolerant behaviour is to minimise the drawbead height in general. Of course, due to manufacturing constraints this is not always possible but should

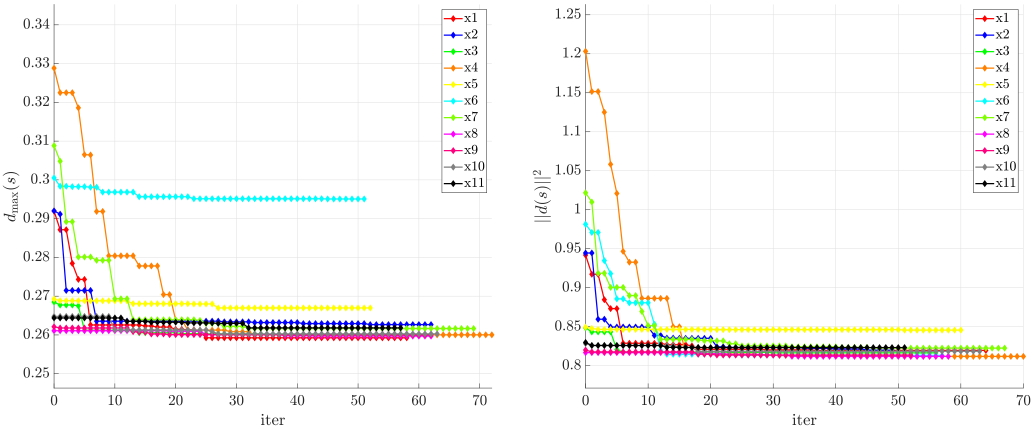


Figure 10.10: Objectives over iterations for both optimisation problems (10.10) and (10.11) of stretch indenting.

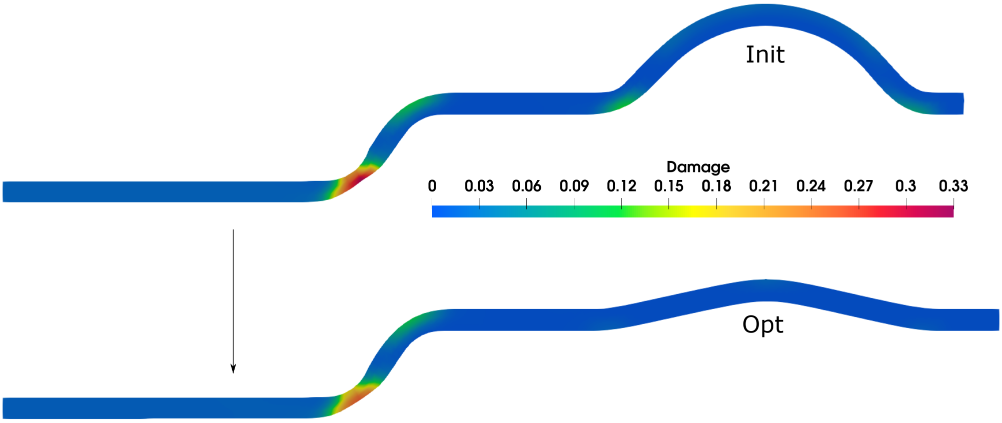


Figure 10.11: Initial and optimal damage distribution due to a change in drawbead design for the $d_{\max}$ optimisation for the initial guess x_4.

be considered.

Extending this research would naturally include the radii r_d and r_p. Since these parameters affect the damage evolution during the process, their inclusion as design variables can benefit the research greatly. This however would significantly increase computational cost. Even with only currently three design variables, optimisation of this process takes a lot of computation time due to the complexity and thereby runtime of each simulation.

Chapter 11

Conclusion of Part II: Industrial challenges

In Chap. 8, the requirements for the approach presented and utilised in Part II: Industrial challenges has been motivated. The necessity and background for optimisation of forming processes has been provided, as this work was funded by the DFG within TRR188. Due to the introduction of contact mechanics for the simulation of forming processes, the method described in Part I: Academic challenges, i.e. an analytical sensitivity analysis, could not easily handle the additional challenges. A brief overview regarding the forming processes optimised in this work, i.e. full forward extrusion, forward hollow extrusion, free air bending, and stretch indenting, has been provided. Necessary quantities, the working parts of each process and influences of process parameters regarding damage have been outlined. Finally, some ideas regarding general optimisation of forming processes has been presented. Two general concepts to optimise forming processes can in theory be pursued. By simplifying a process, contact mechanics can be avoided and optimisation in a classical, *indirect* sense is applied. However, this simplification always results in loss of accuracy in the prediction of the simplified process. Hence, in this work the *direct* optimisation approach, i.e. the process is simulated without these simplifications, has been chosen. While it is computationally more complex, it is more accurate and easier to use in the context of TRR188 with the many forming processes to consider.

Afterwards, in Chap. 9, the optimisation framework Industrial Optimisation Problem Solver (IOPS) using Abaqus for the *direct* optimisation of discontinuous contact problems in forming has been presented. Thereby, the optimisation of forming processes was made possible. The choice for the simulation software has been motivated by its accessibility using Python, and thereby the coupling within an automated optimisation framework. Furthermore its utilisation within TRR188, and hence the ease of cooperation with other projects has been a key factor for using Abaqus. An overview regarding nomenclature of Abaqus files has been given, which eases with understanding of the possible approaches regarding optimisation presented here. The optimisation framework has been described in detail. The beneficial choice of adapting the Abaqus model data base has been highlighted. The framework utilises Abaqus to solve the finite element problem, Python to extract and inject data and Matlab to run the optimisation as a wrapper function. The beneficial flexibility of this approach has been emphasised as it enables optimisation of multiple types of problems simultaneously. Furthermore, this concept allows general process optimisation and thereby is not limited to one type of optimisation problem. This has been proven by the chosen benchmark problems which show possible applications.

Finally, in Chap. 10, several damage optimised forming processes using IOPS have been presented. Bulk metal and sheet metal forming processes have been analysed. Utilisation of the *direct* optimisation approach enables the easy implementation of process constraints and restrictions regarding certain process parameters. This enables the control and reduction of damage in rod extrusion, bending and stretch indenting. All simulations have been computed in Abaqus and contact mechanics have been necessary to accurately represent the forming processes. In contrast to the benchmark optimisations, the problems in this chapter relied heavily on contact interactions and prevented utilisation of SQP. Instead, the gradient-free Nelder-Mead simplex has been applied to optimise these processes.

Results from the numerical optimisation of full forward rod extrusion coincided with data from literature, proving the applicability of the framework to complex forming processes and damage optimisation. Extending the optimisation to the process of forward hollow rod extrusion, enabled initial comparison between both processes. They showed the same characteristics regarding extrusions strain $\varepsilon_{\mathrm{ex}}$ and shoulder angle 2α. The inclusion of the central mandrel radius r_{m} enabled a more complex optimisation of forward hollow extrusion. Inclusion of this variable in the optimisation, and constraining the extrusion strain, yielded an interesting combination of parameters: Neither mandrel radius r_{m}, nor final radius r_1 reached their respective constrained bound and an optimal solution has been generated which lied within a non-intuitive region of the design space.

After the optimisation of the bulk metal forming process for 16MnCrS5, sheet metal forming process for DP800 have been considered. First, elastomer bending has been optimised, by applying shape optimisation to the elastomer cushion. An optimal elastomer type distribution allows a homogenisation of the stress triaxiality over a certain area of the sheet metal. The resulting elastomer shapes might yield a more even damage distribution in a sheet formed this way. The last optimised process has been stretch indenting, a deep-drawing process which prohibits material flow using a drawbead within the die. By optimising the drawbead, the material flow in the process was altered. The geometry has been controlled with three geometric parameters, which acted as design variables of the optimisation. The number of manufacturing constraints highlighted the applicability of the optimisation framework and enabled optimisation of this process without the need to formulate complex replacement constraints. The optimisation with many different initial guesses resulted in a simple, yet efficient statement: To reduce damage in stretch indenting, the drawbead height must be minimised.

Based on the presented work, generalised conclusive remarks can be made from the data, similarly to the results in Part I. These are:

1. Utilisation of numerical optimisation using a *direct* approach may yield parameter combinations which might be unintuitive in nature. In forward hollow extrusion, a combination of parameters was found which lies between constraint bounds and therefore might be missed if normal engineering methods are used to test specific parameter combinations. Furthermore, approximative or simplified methods might omit such oddities in their reduced computations.

2. Generalised statements can be drawn from optimisations, wherein the design variables do not necessary generate one optimised set of variables by themselves, but instead govern other process parameters implicitly, which in turn yield an optimised solution. For stretch indenting for example, the design variables themselves did not directly yield an optimised set, but the underlying conclusion regarding the drawbead height could be drawn.

Chapter 12

Final remarks and outlook

The overall objective of this work was to enable the minimisation and control of damage by applying the concept of mathematical optimisation. In the presented work, it was utilised top optimised damage in may different applications. Initially, to provide a basis for to applied methodology in this work, the concept of mathematical optimisation was briefly outlined. The two main solution strategies, i.e. the gradient-based Sequential Quadratic Programming and the gradient-free Nelder-Mead simplex, were presented. After this general introduction, the work was split in two overarching parts, which tackled different conceptual problems in damage optimisation.

The goal of **Part I: Academic challenges** was to enhance the continuum description for a ductile damage model by means of analytical sensitivity analysis. Therefore, it focused on the problems that arise when a continuum mechanical material model for ductile damage is enhanced with sensitivity data and utilised in optimisation. The underlying material model was able to predict ductile damage, i.e. plasticity coupled to damage, in metal materials. A nonlocal gradient-enhanced formulation was implemented and considered for sensitivity analysis. By using a variational approach, the continuous equations were used to derive the gradient information before discretisation. The geometric description and the constitutive material parameters were chosen as design variables and the resulting gradients derived. They were applied to two different fields of optimisation. In shape optimisation, damage optimised structures were generated. However, the results showed that an unconstrained damage optimisation might yield geometries, which show early localisation of damage. This could be prevented by including damage constraints applied to each finite element (FE) node. Furthermore, a compliance optimisation highlighted the beneficial inclusion of accurate damage modelling, as resulting structures showed a more damage tolerant behaviour. The material sensitivities were applied in a prototype response sensitivity study. Therein, the sensitivity data was directly analysed to give insight into identifiability of material parameters in different stages of experimental loading.

Based on this work, many different fields of research can be followed. Extending the work for the response sensitivity study for example could be combined with optimised shapes for parameter identification (PI). Sensitivity analysis w.r.t. the referential geometry could again be applied to the constitutive material sensitivities. While this a very challenging task, these sensitivities might give further insight on how applicable certain geometries are for PI. Another research direction could be to enhance the underlying material model to incorporate void-closure mechanisms, due to heat treatment for example. This would allow to optimise temperature curves to reduce the accumulation of damage within a pre-damaged problem. Another extension could be the consideration of damage on the microscale and therefore the optimisation of the micromechanical texture of the material under consideration. This in turn could be further enhanced by coupling damage optimisation on the microscale with other forms of optimisation on the macroscale, for example by means of an FE^2 approach [129, 130].

The objective of **Part II: Industrial challenges** was the formulation of an optimisation framework to enable the minimisation and control of damage within the forming processes of TRR188. It presented a methodology to directly optimise these forming processes and thereby problems with discontinuities due to contact mechanics. The necessity to optimise damage in forming process directly was motivated due to the contribution to TRR188. The framework IOPS surrounding the commercial FE software Abaqus was presented. It allowed simultaneous consideration of different typed of design variables which was presented by optimising springback in bending. In addition, it was applied to optimise damage in three different forming processes, i.e. extrusion, bending and stretch indenting. The results showed, that application of numerical optimisation might yield many significant results. On the on hand, the resulting parameter sets might be a combination which is not necessarily analysed in conventional engineering settings. On the other, the results can allow for generalised statements regarding certain parameter combinations, multiple optimal results yield a similar outcome with different design variables sets.

Currently, the presented framework was only applied to 2d-problems. A general enhancement regarding 3d problems is therefore natural. However, this comes with a significant increase in computational complexity. Therefore, either the models have to be setup to improve computation time, or the numerical optimisation has to be adapted. Furthermore, other simplified methods were presented in this thesis. Therefore, comparison between results from direct optimisations and said simplified methods could also be of interest in future work. Finally, another straightforward extension of the presented work could be the addition of more processes within the optimisation. Since the presented framework is not limited forming processes, any type of problems modelled in Abaqus can be subjected to optimisation.

Finally, merging both parts into one concise framework would be of further interest. It would enable optimisation of industrial problems from the academic viewpoint. As was discussed in this work, consideration of contact mechanics for the derivation of gradient information would be required. While this has been proven possible in literature, a robust implementation would be required. Circumventing this necessity could be avoided by utilising surrogate models. This could be structured to enable continuous gradient derivation while simultaneously predicting the desired industrial problem accurately enough. Stating optimisation problems based on these surrogate models then yield solutions which can be applied to the forming process under consideration.

Bibliography

[1] Ahrens, J., B. Geveci, and C. Law: 'ParaView: An End-User Tool for Large Data Visualization'. *Visualization Handbook.* ISBN 978-0123875822. Elesvier, 2005 (cit. on p. 13).

[2] Akeret, R.: 'Failure mechanisms in the bending of aluminum sheets and limits of bendability'. 1978. URL: https://api.semanticscholar.org/CorpusID:136272959 (cit. on p. 105).

[3] Allaire, G.: *Shape Optimization by the Homogenization Method.* Springer-Verlag New York, 2002. DOI: 10.1007/978-1-4684-9286-6 (cit. on p. 4).

[4] Allinger, P. S., N. Bakhtiary, M. Friedrich, O. Müller, F. Mulfinger, M. Puchinger, and J. Sauter: *A New Approach for Sizing, Shape and Toplogy Optimization.* Tech. rep. SAE International, 1996. DOI: 10.4271/960814 (cit. on p. 3).

[5] Allix, O.: 'The bounded rate concept: A framework to deal with objective failure predictions in dynamic within a local constitutive model'. *International Journal of Damage Mechanics* (2013), vol. 22(6): pp. 808–828. DOI: 10.1177/1056789512468355 (cit. on p. 37).

[6] Alzahrani, M., S.-K. Choi, and D. W. Rosen: 'Design of truss-like cellular structures using relative density mapping method'. *Materials & Design* (2015), vol. 85: pp. 349–360. DOI: 10.1016/j.matdes.2015.06.180 (cit. on p. 114).

[7] Anand, R. B., S. K. Shukla, A. Ghorpade, M. K. Tiwari, and R. Shankar: 'Six sigma-based approach to optimize deep drawing operation variables'. *International Journal of Production Research* (2007), vol. 45(10): pp. 2365–2385. DOI: 10.1080/00207540600702308 (cit. on p. 110).

[8] Andrade-Campos, A., S. Coppieters, and M. Strano: 'Optimization and inverse analysis in metal forming: scientific state-of-the-art and recent trends'. *International Journal of Material Forming* (2022), vol. 15(3): p. 44. DOI: 10.1007/s12289-022-01690-8 (cit. on p. 4).

[9] Ansys® Academic Research Mechanical, Release 18.1, Help System, Coupled Field Analysis Guide, ANSYS, Inc: *Ansys.* 2023. URL: http://www.ansys.com (cit. on p. 113).

[10] Atzema, E.: '3 - Formability of auto components'. *Automotive Steels.* Ed. by Rana, R. and S. B. Singh. Woodhead Publishing, 2017: pp. 47–93. doi: 10.1016/B978-0-08-100638-2.00003-1 (cit. on p. 107).

[11] Bahloul, R.: 'Optimisation of process parameters in flanging operation in order to minimise stresses and Lemaitre's damage'. *Materials & Design* (2011), vol. 32(1): pp. 108–120. doi: 10.1016/j.matdes.2010.06.028 (cit. on p. 130).

[12] Baier, H., C. Seesselberg, and B. Specht: *Optimierung in der Strukturmechanik.* Vieweg+Teubner Verlag Wiesbaden, 1994. doi: 10.1007/978-3-322-90700-4 (cit. on p. 3).

[13] Bakhshi-Jooybari, M., M. Saboori, S. Hosseinipour, M. Shakeri, and A. Gorji: 'Experimental and numerical study of optimum die profile in backward rod extrusion'. *Journal of Materials Processing Technology* (2006), vol. 177(1). Proceedings of the 11th International Conference on Metal Forming 2006: pp. 596–599. doi: 10.1016/j.jmatprotec.2006.03.194 (cit. on p. 107).

[14] Barbier, T., E. Shakour, O. Sigmund, G. Lombaert, and M. Schevenels: 'Topology optimization of damage-resistant structures with a predefined load-bearing capacity'. *International Journal for Numerical Methods in Engineering* (2022), vol. 123(4): pp. 1114–1145. doi: 10.1002/nme.6891 (cit. on p. 4).

[16] Barthold, F.-J.: 'Remarks on variational shape sensitivity analysis based on local coordinates'. *Engineering Analysis with Boundary Elements* (2008), vol. 32(11): pp. 971–985. doi: 10.1016/j.enganabound.2007.09.007 (cit. on p. 24).

[17] Barthold, F.-J.: 'Zur Kontinuumsmechanik inverser Geometrieprobleme.' Habilitation. Braunschweiger Schriften zur Mechanik 44, TU Braunschweig, 2002. doi: 10.17877/DE290R-13502 (cit. on pp. 4, 24, 54, 56–58, 62).

[18] Barthold, F.-J. and M. Firuziaan: 'Optimization of hyperelastic materials with isotropic damage'. *Structural and Multidisciplinary Optimization* (2000), vol. 20(1): pp. 12–21. doi: 10.1007/s001580050131 (cit. on p. 56).

[19] Barthold, F.-J. and D. Materna: 'A modified extended finite element method approach for design sensitivity analysis'. *International Journal for Numerical Methods in Engineering* (2015), vol. 104(3): pp. 209–234. doi: 10.1002/nme.4930 (cit. on p. 4).

[15] Barthold, F.-J. and E. Stein: 'A continuum mechanical-based formulation of the variational sensitivity analysis in structural optimization. Part I: analysis'. *Structural optimization* (1996), vol. 11(1): pp. 29–42. doi: 10.1007/BF01279652 (cit. on pp. 54, 56, 57).

[20] Batoz, J.-L., Y. Q. Guo, and F. Mercier: 'The inverse approach with simple triangular shell elements for large strain predictions of sheet metal forming

parts'. *Engineering Compuations* (1998), vol. 15(7): pp. 864–892. DOI: 10.1108/02644409810236894 (cit. on p. 109).

[21] Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind: 'Automatic differentiation in machine learning: a survey'. (2017), vol. 18(1): pp. 5595–5637. DOI: 10.48550/arXiv.1502.05767 (cit. on p. 24).

[22] Bažant, Z. P., T. B. Belytschko, and T.-P. Chang: 'Continuum Theory for Strain-Softening'. *Journal of Engineering Mechanics* (1984), vol. 110(12): pp. 1666–1692. DOI: 10.1061/(ASCE)0733-9399(1984)110:12(1666) (cit. on p. 37).

[23] Bažant, Z. P., T. B. Belytschko, and T.-P. Chang: 'Crack band theory for fracture of concrete'. *Matériaux et Construction* (1983), vol. 16: pp. 155–177. DOI: 10.1007/BF02486267 (cit. on p. 37).

[24] Bažant, Z. P. and G. Pijaudier-Cabot: 'Nonlocal continuum damage, localization instability and convergence'. *Journal of applied mechanics* (1988), vol. 55(2): pp. 287–293. DOI: 10.1115/1.3173674 (cit. on p. 37).

[25] Bažant, Z. P. and M. Jirásek: 'Nonlocal Integral Formulations of Plasticity and Damage: Survey of Progress'. *Journal of Engineering Mechanics* (2002), vol. 128(11): pp. 1119–1149. DOI: 10.1061/(ASCE)0733-9399(2002)128:11(1119) (cit. on p. 37).

[26] Ben Abdessalem, A. and A. El-Hami: 'Global sensitivity analysis and multiobjective optimisation of loading path in tube hydroforming process based on metamodelling techniques'. *The International Journal of Advanced Manufacturing Technology* (2014), vol. 71: pp. 753–773. DOI: 10.1007/s00170-013-5518-4 (cit. on p. 130).

[27] Bendsøe, M.: *Optimization of Structural Topology, Shape, and Material*. Springer Berlin, Heidelberg, 1995. DOI: 10.1007/978-3-662-03115-5 (cit. on pp. 3, 4).

[28] Bendsøe, M. and O. Sigmund: *Topology Optimization*. Springer Berlin, Heidelberg, 2004. DOI: 10.1007/978-3-662-05086-6 (cit. on p. 4).

[29] Bergstra, J. and Y. Bengio: 'Random Search for Hyper-Parameter Optimization'. *Journal of Machine Learning Research* (2012), vol. 13(10): pp. 281–305. URL: http://jmlr.org/papers/v13/bergstra12a.html (cit. on p. 17).

[30] Berthelsen, R., H. Wilbuer, R. Holtermann, and A. Menzel: 'Computational modelling of wear – application to structured surfaces of elastoplastic tools'. *GAMM-Mitteilungen* (2016), vol. 39(2): pp. 210–228. DOI: 10.1002/gamm.201610013 (cit. on p. 114).

[31] Besson, J., C. Berdin, S. Bugat, R. Desmorat, F. Feyel, S. Forest, E. Lorentz, E. Maine, T. Pardoen, A. Pineau, and B. Tanguy: *Local approach to fracture*. Jan. 2004 (cit. on p. 37).

[32] Betten, J.: 'Applications of Tensor Functions in Continuum Damage Mechanics'. *International Journal of Damage Mechanics* (1992), vol. 1(1): pp. 47–59. doi: 10.1177/105678959200100103 (cit. on p. 30).

[33] Birkert, A., S. Haage, and M. Straub: *Umformtechnische Herstellung komplexer Karosserieteile*. Springer Vieweg Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-34670-5 (cit. on p. 145).

[34] Blanchard, P. and E. Brüning: *Direkte Methoden der Variationsrechnung. Ein Lehrbuch*. Springer Vienna, 1982. doi: 10.1007/978-3-7091-2260-0 (cit. on p. 11).

[35] Bletzinger, K.-U., M. Firl, J. Linhard, and R. Wüchner: 'Optimal shapes of mechanically motivated surfaces'. *Computer Methods in Applied Mechanics and Engineering* (2010), vol. 199(5): pp. 324–333. doi: 10.1016/j.cma.2008.09.009 (cit. on p. 4).

[36] Boggs, P. T. and J. W. Tolle: 'Sequential Quadratic Programming'. *Acta Numerica* (1995), vol. 4: pp. 1–51. doi: 10.1017/S0962492900002518 (cit. on p. 21).

[37] Bonet, J. and R. Wood: *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997. doi: 10.1017/CBO9780511755446 (cit. on pp. 30, 55).

[38] Bonte, M. H. A., A. H. van den Boogaard, and J. Huétink: 'An optimisation strategy for industrial metal forming processes'. *Structural and Multidisciplinary Optimization* (2008), vol. 35(6): pp. 571–586. doi: 10.1007/s00158-007-0206-3 (cit. on p. 4).

[39] Box, G. E. P. and N. R. Draper: *Response Surfaces, Mixtures, and Ridge Analyses*. Wiley, 2006. doi: 10.1002/0470072768 (cit. on p. 110).

[40] Brabie, G., E. Costache, N. Nanu, and B. Chirita: 'Prediction and minimisation of sheet thickness variation during deep drawing of micro/milli parts'. *International Journal of Mechanical Sciences* (2013), vol. 68: pp. 277–290. doi: 10.1016/j.ijmecsci.2013.01.028 (cit. on p. 130).

[41] Brepols, T., S. Wulfinghoff, and S. Reese: 'Gradient-extended two-surface damage-plasticity: Micromorphic formulation and numerical aspects'. *International Journal of Plasticity* (2017), vol. 97: pp. 64–106. doi: 10.1016/j.ijplas.2017.05.010 (cit. on p. 30).

[42] Bridgman, P. W.: 'Effects of High Hydrostatic Pressure on the Plastic Properties of Metals'. *Rev. Mod. Phys.* (1 1945), vol. 17: pp. 3–14. doi: 10.1103/RevModPhys.17.3 (cit. on p. 2).

[43] Browne, M. and M. Hillery: 'Optimising the variables when deep-drawing C.R.1 cups'. *Journal of Materials Processing Technology* (2003), vol. 136(1): pp. 64–71. doi: 10.1016/S0924-0136(02)00934-2 (cit. on p. 107).

[44] BROYDEN, C. G.: 'The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations'. *IMA Journal of Applied Mathematics* (1970), vol. 6(1): pp. 76–90. doi: 10.1093/imamat/6.1.76 (cit. on p. 21).

[45] Brünnet, H., N. Lyubenova, M. Müller, J. E. Hoffmann, and D. Bähre: 'Verification and Application of a New 3D Finite Element Approach to Model the Residual Stress Depth Profile after Autofrettage and Consecutive Reaming'. *Procedia CIRP* (2014), vol. 13. 2nd CIRP Conference on Surface Integrity (CSI): pp. 72–77. doi: 10.1016/j.procir.2014.04.013 (cit. on p. 102).

[46] Cao, J., S. Li, Z. Xia, and S. C. Tang: 'Analysis of an axisymmetric deep-drawn part forming using reduced forming steps'. *Journal of Materials Processing Technology* (2001), vol. 117(1): pp. 193–200. doi: 10.1016/S0924-0136(01)01159-1 (cit. on p. 114).

[47] Cao, T. S.: 'Models for ductile damage and fracture prediction in cold bulk metal forming processes: a review'. *International Journal of Material Forming* (2017), vol. 10(2): pp. 139–171. doi: 10.1007/s12289-015-1262-7 (cit. on p. 104).

[48] Chaboche, J.-L.: 'Development of Continuum Damage Mechanics for Elastic Solids Sustaining Anisotropic and Unilateral Damage'. *International Journal of Damage Mechanics* (1993), vol. 2(4): pp. 311–329. doi: 10.1177/105678959300200401 (cit. on p. 30).

[49] Charalampidis, E., P. Kevrekidis, and P. Farrell: 'Computing stationary solutions of the two-dimensional Gross–Pitaevskii equation with deflated continuation'. *Communications in Nonlinear Science and Numerical Simulation* (2018), vol. 54: pp. 482–499. doi: 10.1016/j.cnsns.2017.05.024 (cit. on p. 17).

[50] Choi, K. K. and N. H. Kim: *Structural Sensitivity Analysis and Optimization 1.* Springer-Verlag New York, 2005. doi: 10.1007/b138709 (cit. on p. 10).

[51] Choi, K. K. and N. H. Kim: *Structural Sensitivity Analysis and Optimization 2.* Springer-Verlag New York, 2005. doi: 10.1007/b138895 (cit. on p. 10).

[52] Christensen, P. W. and A. Klarbring: *An Introduction to Structural Optimization.* Springer Dordrecht, 2009. doi: 10.1007/978-1-4020-8666-3 (cit. on p. 4).

[53] Christiansen, S., M. Patriksson, and L. Wynter: ‘Stochastic bilevel programming in structural optimization’. *Structural and Multidisciplinary Optimization* (2001), vol. 21(5): pp. 361–371. doi: 10.1007/s001580100115 (cit. on p. 102).

[54] Coleman, B. D. and W. Noll: ‘The thermodynamics of elastic materials with heat conduction and viscosity’. *Archive for Rational Mechanics and Analysis* (1963), vol. 13(1): pp. 167–178. doi: 10.1007/BF01262690 (cit. on p. 40).

[55] Community, B. O.: *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2020. url: http://www.blender.org (cit. on p. 13).

[56] COMSOL Multiphysics® v. 6.2. COMSOL AB Stockholm, S.: *Comsol*. 2023. url: https://www.comsol.com (cit. on p. 113).

[57] D’Errico, J.: *fminsearchbnd, fminsearchcon: MATLAB Central File Exchange*. 2022. url: https://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon (cit. on pp. 18, 118).

[58] Dassault Systèmes: *Simulia Abaqus Manual*. 2023. url: https://help.3ds.com/2023/English/DSSIMULIA_Established/SIMULIA_Established_FrontmatterMap/sim-r-DSDocAbaqus.htm?contextscope=all (cit. on pp. 13, 113–115, 117, 119, 120, 140).

[59] Debray, K., Z. C. Sun, R. Radjai, Y. Q. Guo, L. Dai, and Y. X. Gu: ‘Optimum Design Of Addendum Surfaces In Sheet Metal Forming Process’. *AIP Conference Proceedings* (2004), vol. 712(1): pp. 1980–1984. doi: 10.1063/1.1766824 (cit. on p. 109).

[60] Desmorat, R., M. Chambart, F. Gatuingt, and D. Guilbaud: ‘Delay-active damage versus non-local enhancement for anisotropic damage dynamics computations with alternated loading’. *Engineering Fracture Mechanics* (2010), vol. 77(12): pp. 2294–2315. doi: 10.1016/j.engfracmech.2010.04.006 (cit. on p. 30).

[61] Dienemann, R., A. Schumacher, and S. Fiebig: ‘Topology optimization for finding shell structures manufactured by deep drawing’. *Structural and Multidisciplinary Optimization* (2017), vol. 56(2): pp. 473–485. doi: 10.1007/s00158-017-1661-0 (cit. on p. 109).

[62] Dienemann, R.: *Entwicklung einer Optimierungsmethodik für die Form- und Topologieoptimierung von tiefziehbaren Blechstrukturen*. Berichte aus dem Maschinenbau. Aachen: Shaker Verlag, 2018 (cit. on p. 109).

[63] Dimitrijevic, B. and K. Hackl: ‘A method for gradient enhancement of continuum damage models’. *Technische Mechanik* (2008), vol. 28(1): pp. 43–52 (cit. on p. 37).

[64] DIN 8580:2022-12: *Manufacturing processes - Terms and definitions, division.* 2022. DOI: 10.31030/3217872 (cit. on p. 102).

[65] Dragon, A. and Z. Mróz: 'A continuum model for plastic-brittle behaviour of rock and concrete'. *International Journal of Engineering Science* (1979), vol. 17(2): pp. 121–137. DOI: 10.1016/0020-7225(79)90058-2 (cit. on p. 2).

[66] Du, C.: 'Micro-plasticity characterization of martensite, ferrite, and dual-phase steel'. English. Proefschrift. Phd Thesis 1 (Research TU/e / Graduation TU/e). Mechanical Engineering, Nov. 2016 (cit. on p. 3).

[67] Duan, L., N.-c. Xiao, G. Li, F. Xu, T. Chen, and A. Cheng: 'Bending analysis and design optimisation of tailor-rolled blank thin-walled structures with top-hat sections'. *International Journal of Crashworthiness* (2017), vol. 22(3): pp. 227–242. DOI: 10.1080/13588265.2016.1246069 (cit. on p. 110).

[68] Duggirala, R., R. Shivpuri, S. Kini, S. Ghosh, and S. Roy: 'Computer aided approach for design and optimization of cold forging sequences for automotive parts'. *Journal of Materials Processing Technology* (1994), vol. 46(1): pp. 185–198. DOI: 10.1016/0924-0136(94)90110-4 (cit. on p. 130).

[69] Duvaut, G., J. L. Lions, C. W. John, and S. C. Cowin: *Inequalities in Mechanics and Physics.* Springer, Berlin, 1976. DOI: 10.1007/978-3-642-66165-5 (cit. on p. 37).

[70] Ekh, M., A. Menzel, K. Runesson, and P. Steinmann: 'Anisotropic damage with the MCR effect coupled to plasticity'. *International Journal of Engineering Science* (2003), vol. 41(13). Damage and failure analysis of materials: pp. 1535–1551. DOI: 10.1016/S0020-7225(03)00032-6 (cit. on p. 30).

[71] Elguedj, T. and T. Hughes: 'Isogeometric analysis of nearly incompressible large strain plasticity'. *Computer Methods in Applied Mechanics and Engineering* (2014), vol. 268: pp. 388–416. DOI: 10.1016/j.cma.2013.09.024 (cit. on pp. 33, 185).

[72] Evans, L. C.: *Partial differential equations.* 1998 (cit. on p. 102).

[73] Faes, M. G. and M. A. Valdebenito: 'Fully decoupled reliability-based design optimization of structural systems subject to uncertain loads'. *Computer Methods in Applied Mechanics and Engineering* (2020), vol. 371: p. 113313. DOI: 10.1016/j.cma.2020.113313 (cit. on p. 17).

[74] Farin, G.: *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, 1993. DOI: 10.1016/C2009-0-22351-8 (cit. on pp. 4, 24, 55, 82).

[75] Farjad Bastani, A., T. Aukrust, and S. Brandal: 'Optimisation of flow balance and isothermal extrusion of aluminium using finite-element simulations'.

Journal of Materials Processing Technology (2011), vol. 211(4): pp. 650–667. DOI: 10.1016/j.jmatprotec.2010.11.021 (cit. on p. 107).

[76] Fischer, A.: 'A special newton-type optimization method'. *Optimization* (1992), vol. 24(3-4): pp. 269–284. URL: https://api.semanticscholar.org/CorpusID:122109163 (cit. on p. 207).

[77] Fishman, G. S.: *Monte Carlo: Concepts, Algorithms, and Applications.* Springer Science+Business Media New York, 1996. DOI: 10.1007/978-1-4757-2553-7 (cit. on p. 17).

[78] Fletcher, R.: 'A new approach to variable metric algorithms'. *The Computer Journal* (Jan. 1970), vol. 13(3): pp. 317–322. DOI: 10.1093/comjnl/13.3.317 (cit. on p. 21).

[79] Fletcher, R.: *Practical methods of optimization.* John Wiley & Sons, 1987 (cit. on p. 21).

[80] Forest, S.: 'Micromorphic Approach for Gradient Elasticity, Viscoplasticity, and Damage'. *Journal of Engineering Mechanics* (2009), vol. 135(3): pp. 117–131. DOI: 10.1061/(ASCE)0733-9399(2009)135:3(117) (cit. on pp. 30, 37).

[81] Forsgren, A., P. E. Gill, and M. H. Wright: 'Interior Methods for Nonlinear Optimization'. *SIAM Review* (2002), vol. 44(4): pp. 525–597. DOI: 10.1137/S0036144502414942 (cit. on p. 20).

[82] Ganapathysubramanian, S. and N. Zabaras: 'Computational design of deformation processes for materials with ductile damage'. *Computer Methods in Applied Mechanics and Engineering* (2003), vol. 192(1): pp. 147–183. DOI: 10.1016/S0045-7825(02)00538-8 (cit. on p. 130).

[83] Garcia-Granada, A. A., J. Catafal-Pedragosa, and H. G. Lemu: 'Topology optimization through stiffness/weight ratio analysis for a three-point bending test of additive manufactured parts'. *IOP Conference Series: Materials Science and Engineering* (2019), vol. 700(1): p. 012012. DOI: 10.1088/1757-899X/700/1/012012 (cit. on p. 108).

[84] Garrison, W. and N. Moody: 'Ductile fracture'. *Journal of Physics and Chemistry of Solids* (1987), vol. 48(11): pp. 1035–1074. DOI: 10.1016/0022-3697(87)90118-1 (cit. on p. 3).

[85] Gelfand, I., S. Fomin, and R. Silverman: *Calculus of Variations.* Dover Publications, New York, 2000 (cit. on p. 11).

[86] Gerzen, N.: 'Analysis and applications of variational sensitivity information in structural optimisation'. PhD Thesis. TU Dortmund University, 2014. DOI: 10.17877/DE290R-919 (cit. on pp. 24, 56).

[87] Gerzen, N., P. Clausen, and C. Pedersen: 'Sizing optimization for industrial applications and best practice design process'. 2016: pp. 41–49. doi: 10.2495/HPSM160041 (cit. on p. 4).

[88] Gharib, H. H., A. S. Wifi, M. Y. Younan, and A. O. Nassef: 'Blank holder force optimisation strategy in deep drawing process'. *International Journal of Computational Materials Science and Surface Engineering* (2007), vol. 1(2): pp. 226–241. doi: 10.1504/IJCMSSE.2007.014873 (cit. on p. 109).

[89] Gibson, I., D. Rosen, B. Stucker, and A. Khorasani: *Additive Manufacturing Technologies*. Nov. 2020. doi: 10.1007/978-3-030-56127-7 (cit. on p. 205).

[90] Gitschel, R., O. Hering, A. Schulze, and A. Erman Tekkaya: 'Controlling Damage Evolution in Geometrically Identical Cold Forged Parts by Counterpressure'. *Journal of Manufacturing Science and Engineering* (Dec. 2022), vol. 145(1). doi: 10.1115/1.4056266 (cit. on pp. 105, 216).

[91] Goldfarb, D.: 'A family of variable-metric methods derived by variational means'. *Mathematics of Computation* (1970), vol. 24(1). url: https://www.ams.org/journals/mcom/1970-24-109/S0025-5718-1970-0258249-6/ (cit. on p. 21).

[92] Gould, N., D. Orband, and P. Toint: 'Numerical methods for large-scale nonlinear optmization'. *Acta Numerica* (2005), vol. 14: pp. 299–361. doi: 10.1017/S0962492904000248 (cit. on p. 20).

[93] Griffiths, C. A., J. Howarth, G. D. Almeida-Rowbotham, and A. Rees: 'A design of experiments approach to optimise tensile and notched bending properties of fused deposition modelling parts'. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* (2016), vol. 230(8): pp. 1502–1512. doi: 10.1177/0954405416640182 (cit. on p. 107).

[94] Gross, D. and T. Seelig: *Fracture Mechanics: With an introduction to Micromechanics*. Springer Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-19240-1 (cit. on p. 104).

[95] Guhr, F. and F.-J. Barthold: 'Damage Optimisation for Air Bending'. *PAMM* (2021), vol. 20(1): e202000074. doi: 10.1002/pamm.202000074 (cit. on pp. 119, 130, 217).

[96] Guhr, F. and F.-J. Barthold: 'Geometric and material sensitivities for elastoplasticity including non-local damage regularisation'. *PAMM* (2023), vol. 23(1): e202200233. doi: 10.1002/pamm.202200233 (cit. on pp. 55, 82).

[97] Guhr, F. and F.-J. Barthold: 'Load Optimisation for Damage Reduction and Fracture Prevention'. *8th GACM Colloquium on Computational Mechanics For*

Young Scientists From Academia and Industry (2019), vol.: pp. 55–58. DOI: 10.17170/kobra-202007161457 (cit. on pp. 117, 119, 130, 217).

[98] Guhr, F. and F.-J. Barthold: ‘Shape optimised geometries for ductile damaging materials’. *PAMM* (2021), vol. 21(1): e202100198. DOI: 10.1002/pamm.202100198 (cit. on pp. 55, 83).

[99] Guhr, F. and F.-J. Barthold: ‘Variational sensitivity analysis and shape optimisation applied to a non-local, ductile damage model’. *Computational Mechanics* (2023), vol. DOI: 10.1007/s00466-023-02377-w (cit. on pp. 4, 10, 55, 82, 186).

[100] Guhr, F., F.-J. Barthold, A. Menzel, L. Sprave, and J. Liedmann: ‘Sensitivity analysis of a non-local, gradient enhanced damage model’. *PAMM* (2018), vol. 18(1): e201800147. DOI: 10.1002/pamm.201800147 (cit. on pp. 55, 82).

[101] Guhr, F., F.-J. Barthold, R. Meya, and A. E. Tekkaya: ‘Load Optimisation for Air Bending in the Context of Damage Reduction’. *PAMM* (2019), vol. 19(1): e201900179. DOI: 10.1002/pamm.201900179 (cit. on pp. 117, 119, 130, 217).

[102] Guhr, F., R. Gitschel, F.-J. Barthold, and A. E. Tekkaya: ‘Numerical optimisation of damage in extrusion processes’. *PAMM* (2023), vol. 23(3): e202300199. DOI: 10.1002/pamm.202300199 (cit. on pp. 129, 130).

[103] Guhr, F., L. Sprave, F.-J. Barthold, and A. Menzel: ‘Computational shape optimisation for a gradient-enhanced continuum damage model’. *Computational Mechanics* (2020), vol. 65(4): pp. 1105–1124. DOI: 10.1007/s00466-019-01810-3 (cit. on pp. 4, 37, 38, 55, 58, 82, 83, 88, 205–207).

[104] Gupta, A. K.: ‘Predictive modelling of turning operations using response surface methodology, artificial neural networks and support vector regression’. *International Journal of Production Research* (2010), vol. 48(3): pp. 763–778. DOI: 10.1080/00207540802452132 (cit. on p. 110).

[105] Gurson, A. L.: ‘Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Media’. *Journal of Engineering Materials and Technology* (1977), vol. 99(1): pp. 2–15. DOI: 10.1115/1.3443401 (cit. on p. 30).

[106] Hahn, M. and A. E. Tekkaya: ‘Part-optimized forming by spatially distributed vaporizing foil actuators’. *International Journal of Material Forming* (2021), vol. 14(6): pp. 1391–1401. DOI: 10.1007/s12289-021-01634-8 (cit. on p. 114).

[107] Harris, C. R. et al.: ‘Array programming with NumPy’. *Nature* (Sept. 2020), vol. 585(7825): pp. 357–362. DOI: 10.1038/s41586-020-2649-2 (cit. on pp. 13, 122).

[108] Hartmann, S. and R. R. Gilbert: 'Material parameter identification using finite elements with time-adaptive higher-order time integration and experimental full-field strain information'. *Computational Mechanics* (2021), vol. 68(3): pp. 633–650. doi: 10.1007/s00466-021-01998-3 (cit. on p. 4).

[109] Hering, O.: 'Schädigung in der Kaltmassivumformung: Entwicklung, Auswirkungen und Kontrolle'. PhD thesis. TU University Dortmund, 2020. doi: 10.17877/de290r-21691 (cit. on pp. 104, 130).

[110] Hering, O., A. Dunlap, A. E. Tekkaya, A. Aretz, and A. Schwedt: 'Characterization of damage in forward rod extruded parts'. *International Journal of Material Forming* (Nov. 2020), vol. 13(6): pp. 1003–1014. doi: 10.1007/s12289-019-01525-z (cit. on pp. 104, 130, 216).

[111] Hering, O. and A. E. Tekkaya: 'Damage-induced performance variations of cold forged parts'. *Journal of Materials Processing Technology* (2020), vol. 279: p. 116556. doi: 10.1016/j.jmatprotec.2019.116556 (cit. on pp. 104, 129, 132, 134).

[112] Hill, R.: 'Aspects of Invariance in Solid Mechanics'. Ed. by Yih, C.-S. Vol. 18. Advances in Applied Mechanics. Elsevier, 1978: pp. 1–75. doi: 10.1016/S0065-2156(08)70264-3 (cit. on p. 32).

[113] Hiriart-Urruty, J.-B. and C. Lemaréchal: *Convex Analysis and Minimization Algorithms I*. Springer-Verlag Berlin Heidelberg, 1993. doi: 10.1007/978-3-662-02796-7 (cit. on pp. 16, 102).

[114] Hirt, G., A. E. Tekkaya, T. Clausmeyer, and J. Lohmar: 'Potential and status of damage controlled forming processes'. *Production Engineering* (2020), vol. 14(1): pp. 1–4. doi: 10.1007/s11740-019-00948-6 (cit. on p. 3).

[115] Hoefnagels, J. P. M., C. C. Tasan, F. Maresca, F. J. Peters, and V. G. Kouznetsova: 'Retardation of plastic instability via damage-enabled microstrain delocalization'. *Journal of Materials Science* (2015), vol. 50(21): pp. 6882–6897. doi: 10.1007/s10853-015-9164-0 (cit. on p. 3).

[116] Hoppe, R. H. W. and S. I. Petrova: 'Primal–dual Newton interior point methods in shape and topology optimization'. *Numerical Linear Algebra with Applications* (2004), vol. 11(5-6): pp. 413–429. doi: 10.1002/nla.353 (cit. on p. 20).

[117] Horn, R. A.: *Topics in Matrix Analysis*. Cambridge University Press, 1991. doi: 10.1017/CBO9780511840371 (cit. on p. 190).

[118] Hosford, W. and R. Caddell: *Metal Forming: Mechanics and Metallurgy*. Cambridge University Press, 2011. url: https://books.google.de/books?id=x0GWAd7hNy0C (cit. on p. 106).

[119] Hu, Z.: 'Realisation and application of size dependent FEM-simulation for deep drawing of rectangular work pieces'. *CIRP Journal of Manufacturing Science and Technology* (2011), vol. 4(1): pp. 90–95. doi: 10.1016/j.cirpj.2011.05.006 (cit. on p. 107).

[120] Hughes, T. J. R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000 (cit. on pp. 30, 55).

[121] Huh, H. and S.-H. Kim: 'Optimum Process Design in Sheet-Metal Forming With Finite Element Analysis '. *Journal of Engineering Materials and Technology* (2000), vol. 123(4): pp. 476–481. doi: 10.1115/1.1395579 (cit. on p. 111).

[122] Jin, W. and C. Arson: 'Anisotropic nonlocal damage model for materials with intrinsic transverse isotropy'. *International Journal of Solids and Structures* (2018), vol. 139-140: pp. 29–42. doi: 10.1016/j.ijsolstr.2018.01.020 (cit. on p. 30).

[123] Jirásek, M. and M. Bauer: 'Numerical aspects of the crack band approach'. *Computers & Structures* (2012), vol. 110-111: pp. 60–78. doi: 10.1016/j.compstruc.2012.06.006 (cit. on p. 37).

[124] Kachanov., L.: 'Time of rupture process under creep conditions (in russian)'. *Izvestia Akademii Nauk, USSR* (1958), vol. 8: pp. 26–31 (cit. on pp. 2, 30).

[125] Kamat, M. P.: 'Structural optimization: Status and promise'. *NASA STI/Recon Technical Report A* (1993), vol. 93: p. 30075. doi: 10.2514/4.866234 (cit. on p. 4).

[126] Kazymyrovych, V.: 'Very high cycle fatigue of engineering materials : A literature review'. *Karlstad University* (2009), vol. url: https://urn.kb.se/resolve?urn=urn:nbn:se:kau:diva-3933 (cit. on p. 2).

[127] Kick, M. and P. Junker: 'A novel approach for the consideration of plastic material behavior in thermodynamic topology optimization'. *PAMM* (2021), vol. 21(1): e202100075. doi: 10.1002/pamm.202100075 (cit. on p. 4).

[128] Kiefer, B., T. Waffenschmidt, L. Sprave, and A. Menzel: 'A gradient-enhanced damage model coupled to plasticity—multi-surface formulation and algorithmic concepts'. *International Journal of Damage Mechanics* (2018), vol. 27(2): pp. 253–295. doi: 10.1177/1056789516676306 (cit. on p. 47).

[129] Kijanski, W.: *Optimal Material Design based on Variational Sensitivity Analysis*. Shaker Verlag, 2019 (cit. on pp. 4, 11, 56, 60, 152).

[130] Kijanski, W. and F.-J. Barthold: 'Two-scale shape optimisation based on numerical homogenisation techniques and variational sensitivity analysis'. *Com-*

putational Mechanics (2021), vol. 67(4): pp. 1021–1040. DOI: 10.1007/s00466-020-01955-6 (cit. on pp. 4, 24, 57, 152).

[131] Kim, N. H., K. K. Choi, and J. S. Chen: 'Die shape design optimization of sheet metal stamping process using meshfree method'. *International Journal for Numerical Methods in Engineering* (2001), vol. 51(12): pp. 1385–1405. DOI: 10.1002/nme.181 (cit. on p. 111).

[132] Klarbring, A. and G. Bjöourkman: 'Solution of large displacement contact problems with friction using Newton's method for generalized equations'. *International Journal for Numerical Methods in Engineering* (1992), vol. 34(1): pp. 249–269. DOI: 10.1002/nme.1620340116 (cit. on p. 101).

[133] Kleinermann, J.-P. and J.-P. Ponthot: 'Parameter identification and shape/process optimization in metal forming simulation'. *Journal of Materials Processing Technology* (2003), vol. 139(1). IMCC2000 Vol. 2 S.I.: pp. 521–526. DOI: 10.1016/S0924-0136(03)00530-2 (cit. on p. 111).

[134] Kleuter, B., A. Menzel, and P. Steinmann: 'Generalized parameter identification for finite viscoelasticity'. *Computer Methods in Applied Mechanics and Engineering* (2007), vol. 196(35): pp. 3315–3334. DOI: 10.1016/j.cma.2007.03.010 (cit. on p. 4).

[135] Klocke, F. and W. König: *Fertigungsverfahren 4*. Springer-Verlag, Berlin, Heidelberg, 2006. DOI: 10.1007/978-3-540-39533-1 (cit. on pp. 105, 106).

[136] Kristiansen, H., K. Poulios, and N. Aage: 'Topology optimization for compliance and contact pressure distribution in structural problems with friction'. *Computer Methods in Applied Mechanics and Engineering* (2020), vol. 364: p. 112915. DOI: 10.1016/j.cma.2020.112915 (cit. on p. 101).

[137] Landron, C., O. Bouaziz, E. Maire, and J. Adrien: 'Characterization and modeling of void nucleation by interface decohesion in dual phase steels'. *Scripta Materialia* (2010), vol. 63(10): pp. 973–976. DOI: 10.1016/j.scriptamat.2010.07.021 (cit. on p. 3).

[138] Lange, K.: *Umformtechnik: Handbuch für Industrie und Wissenschaft*. Springer Berlin, Heidelberg, 1990. DOI: 10.1007/978-3-662-10686-0 (cit. on pp. 105, 106).

[139] Lange, K., M. Kammerer, K. Pöhlandt, and J. Schöck: *Fließpressen: Wirtschaftliche Fertigung metallischer Präzisionswerkstücke*. Springer Berlin, Heidelberg, 2008. DOI: 10.1007/978-3-540-30910-9 (cit. on p. 105).

[140] Langenfeld, K., P. Junker, and J. Mosler: 'Quasi-brittle damage modeling based on incremental energy relaxation combined with a viscous-type regularization'.

Continuum Mechanics and Thermodynamics (2018), vol. 30(5): pp. 1125–1144. DOI: 10.1007/s00161-018-0669-z (cit. on p. 37).

[141] LEMAITRE, J.: *A Course on Damage Mechanics*. Springer-Verlag, 1996. DOI: 10.1007/978-3-642-18255-6 (cit. on p. 30).

[142] LEMAITRE, J. and J. DUFAILLY: 'Damage measurements'. *Engineering Fracture Mechanics* (1987), vol. 28(5). Special Issue in Honor of Professor Takeo Yokobori: pp. 643–661. DOI: 10.1016/0013-7944(87)90059-2 (cit. on p. 30).

[143] LEVENBERG, K.: 'A Method for the Solution of Certain Problems in Least Squares'. *Quart. Appl. Math. 2* (1944), vol. 2(2): pp. 164–168. URL: https://www.jstor.org/stable/43633451 (cit. on p. 20).

[144] LIEBE, T., P. STEINMANN, and A. BENALLAL: 'Theoretical and computational aspects of a thermodynamically consistent framework for geometrically linear gradient damage'. *Computer Methods in Applied Mechanics and Engineering* (2001), vol. 190(49): pp. 6555–6576. DOI: 10.1016/S0045-7825(01)00250-X (cit. on pp. 36–38).

[145] LIEDMANN, J.: 'Elastic-Plastic Design Sensitivities based on Variational Analysis and Applications in Optimal Specimen Design'. PhD thesis. TU University Dortmund, 2021. DOI: 10.17877/DE290R-22420 (cit. on pp. 4, 23, 45, 54–56, 60, 130, 186, 205).

[146] LIEDMANN, J. and F.-J. BARTHOLD: 'Variational sensitivity analysis of elastoplastic structures applied to optimal shape of specimens'. *Structural and Multidisciplinary Optimization* (2020), vol. 61(6): pp. 2237–2251. DOI: 10.1007/s00158-020-02492-9 (cit. on pp. 4, 24, 54, 58, 60, 186, 205).

[147] LIEDMANN, J., S. GERKE, F.-J. BARTHOLD, and M. BRÜNIG: 'Shape optimization of the X0-specimen: theory, numerical simulation and experimental verification'. *Computational Mechanics* (2020), vol. 66(6): pp. 1275–1291. DOI: 10.1007/s00466-020-01900-7 (cit. on pp. 54, 55, 60, 186, 205).

[148] LIEDMANN, J. and F.-J. BARTHOLD: 'Sensitivity Analysis of Nonlinear Structural Response regarding Geometry and External Loads'. *PAMM* (2018), vol. 18(1): e201800135. DOI: 10.1002/pamm.201800135 (cit. on p. 217).

[149] LOVELOCK, D. and H. RUND: *Tensors, Differential Forms, and Variational Principles*. Dover Books on Mathematics. Dover Publications, New York, 1989 (cit. on p. 11).

[150] LUDWIK, P.: 'A unified approach for parameter identification of inelastic material models in the frame of the finite element method'. *Zeitschrift für Metallkunde* (1926), vol. 18(9): pp. 269–272 (cit. on p. 2).

[151] Luenberger, D. G. and Y. Ye: *Linear and Nonlinear Programming.* Springer Cham, 2016. doi: 10.1007/978-3-319-18842-3 (cit. on p. 47).

[152] Mahnken, R.: 'Theoretische und numerische Aspekte zur Modellierung und Parameteridentikation bei metallischen Werkstoffen'. Habilitation. Universität Hannover, 1996 (cit. on p. 24).

[153] Mahnken, R. and E. Stein: 'A unified approach for parameter identification of inelastic material models in the frame of the finite element method'. *Computer Methods in Applied Mechanics and Engineering* (1996), vol. 136(3): pp. 225–258. doi: 10.1016/0045-7825(96)00991-7 (cit. on p. 4).

[154] Marquardt, D.: 'An Algorithm for Least-Squares Estimation of Nonlinear Parameters'. *SIAM J. Appl. Math.* (1963), vol. 11(2): pp. 431–441. doi: 10.1137/0111030 (cit. on p. 20).

[155] Materna, D.: 'Structural and Sensitivity Analysis for the Primal and Dual Problems in the Physical and Material Spaces'. PhD thesis. TU University Dortmund, 2009. doi: 10.17877/DE290R-474 (cit. on pp. 11, 62).

[156] MATLAB: *version 9.12.0 (R2022a).* Natick, Massachusetts, United States: The MathWorks Inc., 2022. url: https://www.mathworks.com (cit. on pp. 12, 21, 117).

[157] Menna, C. and L. Esposito: 'Flexural Behaviour of Steel-Reinforced Topology-Optimised Beams Fabricated by 3D Concrete Printing'. *Third RILEM International Conference on Concrete and Digital Fabrication.* Ed. by Buswell, R., A. Blanco, S. Cavalaro, and P. Kinnell. Springer International Publishing, 2022: pp. 404–410. doi: 10.1007/978-3-031-06116-5_60 (cit. on p. 108).

[158] Menzel, A. and P. Steinmann: 'A theoretical and computational framework for anisotropic continuum damage mechanics at large strains'. *International Journal of Solids and Structures* (2001), vol. 38(52): pp. 9505–9523. doi: 10.1016/S0020-7683(01)00136-6 (cit. on p. 30).

[159] Metropolis, N.: 'The Beginning of the Monte Carlo Method'. *Los Alamos Science Special Issue* (1987), vol. 15: pp. 125–130 (cit. on p. 17).

[160] Meya, R.: 'Schädigungskontrolliertes Blechbiegen mittels Druckspannungsüberlagerung'. PhD thesis. TU University Dortmund, 2020. doi: 10.17877/DE290R-21688 (cit. on pp. 105, 130, 139).

[161] Meya, R., C. Löbbe, and A. E. Tekkaya: 'Stress State Control by a Novel Bending Process and its Effect on Damage and Product Performance'. *Journal of Manufacturing Science and Engineering* (2019), vol. 141(10): p. 101013. doi: 10.1115/1.4044394 (cit. on pp. 105, 129, 139, 222).

[162] Miehe, C. and J. Schröder: ‘A comparative study of stress update algorithms for rate-independent and rate-dependent crystal plasticity’. *International Journal for Numerical Methods in Engineering* (2001), vol. 50(2): pp. 273–298. doi: 10.1002/1097-0207(20010120)50:2<273::AID-NME17>3.0.CO;2-Q (cit. on p. 47).

[163] Mkaddem, A. and R. Bahloul: ‘Experimental and numerical optimisation of the sheet products geometry using response surface methodology’. *Journal of Materials Processing Technology* (2007), vol. 189(1): pp. 441–449. doi: 10.1016/j.jmatprotec.2007.02.026 (cit. on p. 130).

[164] Müller, M., T. Herrig, and T. Bergs: ‘Numerical analysis of the load paths and the resulting damage evolution during deep drawing of dual-phase steel’. *IOP Conference Series: Materials Science and Engineering* (2023), vol. doi: 10.1088/1757-899X/1284/1/012006 (cit. on p. 106).

[165] Müller, M., I. Weiser, T. Herrig, and T. Bergs: ‘Influence of process parameters and process set-up on damage evolution during stretch drawing of u-shaped profiles’. *Materials Research Proceedings* (2023), vol. 28: pp. 727–736. doi: 10.21741/9781644902479-79 (cit. on pp. 106, 144).

[166] Müller, M., I. F. Weiser, T. Herrig, and T. Bergs: ‘Numerical Prediction of the Influence of Process Parameters and Process Set-Up on Damage Evolution during Deep Drawing of Rectangular Cups’. *Engineering Proceedings* (2022), vol. 26(1). doi: 10.3390/engproc2022026006 (cit. on pp. 106, 144).

[167] Murakami, S.: *Continuum Damage Mechanics: A Continuum Mechanics Approach to the Analysis of Damage and Fracture (Solid Mechanics and Its Applications)*. Springer, 2012. doi: 10.1007/978-94-007-2666-6 (cit. on p. 30).

[168] Naceur, H., A. Delaméziere, J. Batoz, Y. Guo, and C. Knopf-Lenoir: ‘Some improvements on the optimum process design in deep drawing using the inverse approach’. *Journal of Materials Processing Technology* (2004), vol. 146(2): pp. 250–262. doi: 10.1016/j.jmatprotec.2003.11.015 (cit. on p. 109).

[169] Neidinger, R. D.: ‘Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming’. *SIAM Review* (2010), vol. 52(3): pp. 545–563. doi: 10.1137/080743627 (cit. on p. 24).

[170] Nelder, J. A. and R. Mead: ‘A Simplex Method for Function Minimization’. *The Computer Journal* (1965), vol. 7(4): pp. 308–313. doi: 10.1093/comjnl/7.4.308 (cit. on p. 17).

[171] Nick, M., A. Feuerhack, T. Bergs, and T. Clausmeyer: ‘Numerical Investigation of Damage in Single-step, Two-step, and Reverse Deep Drawing of Rotationally Symmetric Cups from DP800 Dual Phase Steel’. *Procedia Manufacturing* (2020),

vol. 47. 23rd International Conference on Material Forming: pp. 636–642. DOI: 10.1016/j.promfg.2020.04.195 (cit. on p. 106).

[172] Nick, M., R. Mannens, D. Trauth, and T. Bergs: ‘Load path modelling in single-step deep drawing of rotationally symmetric cups’. *Procedia Manufacturing* (2019), vol. 29. “18th International Conference on Sheet Metal, SHEMET 2019”“New Trends and Developments in Sheet Metal Processing”: pp. 520–527. DOI: 10.1016/j.promfg.2019.02.170 (cit. on pp. 106, 129).

[173] Niehüser, A. and J. Mosler: ‘Numerically efficient and robust interior-point algorithm for finite strain rate-independent crystal plasticity’. *Computer Methods in Applied Mechanics and Engineering* (2023), vol. 416: p. 116392. DOI: 10.1016/j.cma.2023.116392 (cit. on p. 20).

[174] Nocedal, J. and S. J. Wright: *Numerical Optimization.* Springer New York, NY, 1999. DOI: 10.1007/978-0-387-40065-5 (cit. on pp. 16, 21, 23, 117).

[175] Noel, L., P. Duysinx, and K. Maute: ‘Level set topology optimization considering damage’. *Structural and Multidisciplinary Optimization* (2017), vol. 56(4): pp. 737–753. DOI: 10.1007/s00158-017-1724-2 (cit. on p. 4).

[176] Okorokov, V., M. Morgantini, Y. Gorash, T. Comlekci, D. Mackenzie, and R. van Rijswick: ‘Corrosion Fatigue of Low Carbon Steel under Compressive Residual Stress Field’. *Procedia Engineering* (2018), vol. 213. 7th International Conference on Fatigue Design, Fatigue Design 2017, 29-30 November 2017, Senlis, France: pp. 674–681. DOI: 10.1016/j.proeng.2018.02.063 (cit. on p. 102).

[177] Ostwald, R., E. Kuhl, and A. Menzel: ‘On the implementation of finite deformation gradient-enhanced damage models’. *Computational Mechanics* (2019), vol. 64(3): pp. 847–877. DOI: 10.1007/s00466-019-01684-5 (cit. on p. 115).

[178] Papadopoulos, I. P. A., P. E. Farrell, and T. M. Surowiec: ‘Computing Multiple Solutions of Topology Optimization Problems’. *SIAM Journal on Scientific Computing* (2021), vol. 43(3): A1555–A1582. DOI: 10.1137/20M1326209 (cit. on p. 17).

[179] Patrick E. Farrell, M. C. and T. M. Surowiec: ‘Deflation for semismooth equations’. *Optimization Methods and Software* (2020), vol. 35(6): pp. 1248–1271. DOI: 10.1080/10556788.2019.1613655 (cit. on p. 17).

[180] Peerlings, R. H. J., R. de Borst, W. A. M. Brekelmans, J. H. P. de Vree, and I. Spee: ‘Some observations on localisation in non-local and gradient damage models’. *European Journal of Mechanics. A, Solids* (1996), vol. 15: pp. 937–953 (cit. on p. 37).

[181] Peerlings, R., M. Geers, R. de Borst, and W. Brekelmans: ‘A critical comparison of nonlocal and gradient-enhanced softening continua’. *International*

Journal of Solids and Structures (2001), vol. 38(44): pp. 7723–7746. DOI: 10.1016/S0020-7683(01)00087-7 (cit. on p. 37).

[182] PETERSON, J. and M. PATRIKSSON: ‘Topology optimization of sheets in contact by a subgradient method’. *International Journal for Numerical Methods in Engineering* (1997), vol. 40(7): pp. 1295–1321. DOI: 10.1002/(SICI)1097-0207(19970415)40:7<1295::AID-NME115>3.0.CO;2-P (cit. on p. 102).

[183] POWELL, M. J. D.: ‘A fast algorithm for nonlinearly constrained optimization calculations’. *Numerical Analysis.* Ed. by WATSON, G. A. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978: pp. 144–157 (cit. on p. 21).

[184] RASHIDA, R. A. R., S. H. MASOODA, D. RUANA, S. PALANISAMYA, X. HUANGA, and R. A. R. RASHIDA: ‘Topology Optimisation of Additively Manufactured Lattice beams for three-point bending test’. 2018. URL: https://api.semanticscholar.org/CorpusID:209373847 (cit. on p. 108).

[185] REMMERS, W.: ‘Causes of cuppy wire’. *Transactions of the Metallurigcal Society of AIME* (1930), vol. 89: pp. 107–120 (cit. on p. 2).

[186] REUSCH, F.: ‘Entwicklung und Anwendung eines nicht-lokalen Materialmodells zur Simulation duktiler Schädigung in metallischen Werkstoffen’. PhD thesis. TU University Dortmund, 2003. DOI: 10.17877/DE290R-16074 (cit. on p. 3).

[187] RIEGER, A.: ‘On the parameter identification of complex material models based on real and virtual experimental data’. PhD thesis. University of Stuttgart, 2022. DOI: 10.18419/opus-199 (cit. on p. 25).

[188] ROGERS, H.: ‘The tensile fracture of ductile metals’. *Transactions of the metallurgical society of AIME* (1960), vol. 218: pp. 498–506 (cit. on p. 30).

[189] ROJAS-LABANDA, S. and M. STOLPE: ‘Benchmarking optimization solvers for structural topology optimizationmetals’. *Structural and Multidisciplinary Optimization* (2015), vol. 52(3): pp. 527–547. DOI: 10.1007/s00158-015-1250-z (cit. on p. 20).

[190] ROSE, L.: ‘Optimisation based parameter identification using optical field measurements’. PhD thesis. TU University Dortmund, 2022. DOI: 10.17877/DE290R-22597 (cit. on pp. 4, 24, 72).

[191] ROSE, L. and A. MENZEL: ‘Identification of thermal material parameters for thermomechanically coupled material models’. *Meccanica* (2021), vol. 56(2): pp. 393–416. DOI: 10.1007/s11012-020-01267-2 (cit. on p. 25).

[192] ROSE, L. and A. MENZEL: ‘Optimisation based material parameter identification using full field displacement and temperature measurements’. *Mechanics of Ma-*

terials (2020), vol. 145: p. 103292. DOI: 10.1016/j.mechmat.2019.103292 (cit. on pp. 4, 25, 72).

[193] Schiefenbusch, J.: ‘Untersuchungen zur Verbesserung des Umformverhaltens von Blechen beim Biegen’. PhD thesis. Universität Dortmund, 1983 (cit. on p. 105).

[194] Schittkowski, K.: *NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems.* 1985 (cit. on pp. 16, 20, 21).

[195] Schlüter, N., F. Grimpe, W. Bleck, and W. Dahl: ‘Modelling of the damage in ductile steels’. *Computational Materials Science* (1996), vol. 7(1). Zeolites and related materials: Trends, targets and challenges: pp. 27–33. DOI: 10.1016/S0927-0256(96)00056-0 (cit. on p. 3).

[196] Schmidt-Baldassari, M.: ‘Numerical concepts for rate-independent single crystal plasticity’. *Computer Methods in Applied Mechanics and Engineering* (2003), vol. 192(11): pp. 1261–1280. DOI: 10.1016/S0045-7825(02)00563-7 (cit. on p. 207).

[197] Schowtjak, A.: ‘Vorhersage der Porenentwicklung in der Umformtechnik’. PhD thesis. Technische Universität Dortmund, 2023 (cit. on p. 114).

[198] Schowtjak, A., R. Schulte, T. Clausmeyer, R. Ostwald, A. E. Tekkaya, and A. Menzel: ‘ADAPT — A Diversely Applicable Parameter Identification Tool: Overview and full-field application examples’. *International Journal of Mechanical Sciences* (2022), vol. 213: p. 106840. DOI: 10.1016/j.ijmecsci.2021.106840 (cit. on pp. 114, 119).

[199] Schumacher, A.: *Optimierung mechanischer Strukturen: Grundlagen und industrielle Anwendungen.* Springer Berlin, Heidelberg, 2005. DOI: 10.1007/b138046 (cit. on p. 3).

[200] Seth, B. R.: ‘Generalized Strain Measure with Applications to Physical Problems’. *Technical report, Wisconsin Univ-Madison Mathematics Research Center* (1961), vol.: pp. 162–172 (cit. on p. 32).

[201] Shanno, D. F.: ‘Conditioning of Quasi-Newton Methods for Function Minimization’. *Mathematics of Computation* (1970), vol. 24: pp. 647–656. URL: https://www.ams.org/journals/mcom/1970-24-111/S0025-5718-1970-0274029-X/ (cit. on p. 21).

[202] Sigmund, O. and S. Torquato: ‘Design of materials with extreme thermal expansion using a three-phase topology optimization method’. *Journal of the Mechanics and Physics of Solids* (1997), vol. 45(6): pp. 1037–1067. DOI: 10.1016/S0022-5096(96)00114-7 (cit. on p. 4).

[203] SIGMUND, O.: ‘A 99 line topology optimization code written in Matlab’. *Structural and Multidisciplinary Optimization* (2001), vol. 21(2): pp. 120–127. DOI: 10.1007/s001580050176 (cit. on p. 4).

[204] SIGMUND, O. and K. MAUTE: ‘Topology optimization approaches’. *Structural and Multidisciplinary Optimization* (2013), vol. 48(6): pp. 1031–1055. DOI: 10.1007/s00158-013-0978-6 (cit. on p. 4).

[205] SIMO, J. C. and T. J. R. HUGHES: *Computational Inelasticity.* Springer New York, NY, 2000. DOI: 10.1007/b98904 (cit. on p. 45).

[206] SIMO, J.: ‘A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part I. Continuum formulation’. *Computer Methods in Applied Mechanics and Engineering* (1988), vol. 66(2): pp. 199–219. DOI: 10.1016/0045-7825(88)90076-X (cit. on pp. 33, 47).

[207] SKY, A.: ‘Higher order finite elements for relaxed micromorphic continua’. PhD thesis. TU Dortmund University, 2022. DOI: 10.17877/DE290R-22996 (cit. on p. 24).

[208] SOKOŁOWSKI, J. and J. ZOLÉSIO: *Introduction to shape optimization: shape sensitivity analysis.* Springer-Verlag Berlin, Heidelberg, 1992. DOI: 10.1007/978-3-642-58106-9 (cit. on pp. 4, 10).

[209] SOUSA, L., C. CASTRO, and C. ANTÓNIO: ‘Optimal design of V and U bending processes using genetic algorithms’. *Journal of Materials Processing Technology* (2006), vol. 172(1): pp. 35–41. DOI: 10.1016/j.jmatprotec.2005.08.011 (cit. on p. 114).

[210] SOUZA NETO, de, D. R. J. OWEN, and D. PERIC: *Computational Methods for Plasticity: Theory and Applications.* Wiley, New York, 2008. DOI: 10.1002/9780470694626 (cit. on pp. 33, 45, 185).

[211] SOYARSLAN, C. and A. E. TEKKAYA: ‘Finite deformation plasticity coupled with isotropic damage: Formulation in principal axes and applications’. *Finite Elements in Analysis and Design* (2010), vol. 46(8): pp. 668–683. DOI: 10.1016/j.finel.2010.03.006 (cit. on p. 107).

[212] SOYARSLAN, C., M. MALEKIPOUR GHARBI, and A. TEKKAYA: ‘A combined experimental–numerical investigation of ductile fracture in bending of a class of ferritic–martensitic steel’. *International Journal of Solids and Structures* (2012), vol. 49(13): pp. 1608–1626. DOI: 10.1016/j.ijsolstr.2012.03.009 (cit. on p. 107).

[213] SPRAVE, L. and A. MENZEL: ‘Regularisation of gradient-enhanced damage coupled to Finite Plasticity.’ *Proceedings of the 7th GACM Colloquium on Computational*

Mechanics, Stuttgart, Germany. 2017: pp. 769–772. DOI: 10.18419/opus-9334 (cit. on pp. 82, 185).

[214] Sprave, L. and A. Menzel: ‘A large strain anisotropic ductile damage model — Effective driving forces and gradient-enhancement of damage vs. plasticity’. *Computer Methods in Applied Mechanics and Engineering* (2023), vol. 416: p. 116284. DOI: 10.1016/j.cma.2023.116284 (cit. on p. 30).

[215] Sprave, L. and A. Menzel: ‘A large strain gradient-enhanced ductile damage model: finite element formulation, experiment and parameter identification’. *Acta Mechanica* (2020), vol. 231(12): pp. 5159–5192. DOI: 10.1007/s00707-020-02786-5 (cit. on pp. 25, 30, 38, 41, 42, 72, 81, 82, 92, 93, 95, 183, 186, 187).

[216] Sprave, L., A. Schowtjak, R. Meya, T. Clausmeyer, A. E. Tekkaya, and A. Menzel: ‘On mesh dependencies in finite-element-based damage prediction: application to sheet metal bending’. *Production Engineering* (2020), vol. 14(1): pp. 123–134. DOI: 10.1007/s11740-019-00937-9 (cit. on p. 144).

[217] Srikanth, A. and N. Zabaras: ‘Shape optimization and preform design in metal forming processes’. *Computer Methods in Applied Mechanics and Engineering* (2000), vol. 190(13): pp. 1859–1901. DOI: 10.1016/S0045-7825(00)00213-9 (cit. on p. 101).

[218] Tanaka, M., M. Fujikawa, D. Balzani, and J. Schröder: ‘Robust numerical calculation of tangent moduli at finite strains based on complex-step derivative approximation and its application to localization analysis’. *Computer Methods in Applied Mechanics and Engineering* (2014), vol. 269: pp. 454–470. DOI: 10.1016/j.cma.2013.11.005 (cit. on p. 23).

[219] Tasan, C., M. Diehl, D. Yan, C. Zambaldi, P. Shanthraj, F. Roters, and D. Raabe: ‘Integrated experimental–simulation analysis of stress and strain partitioning in multiphase alloys’. *Acta Materialia* (2014), vol. 81: pp. 386–400. DOI: 10.1016/j.actamat.2014.07.071 (cit. on p. 3).

[220] Taylor, R. and S. Govindjee: *FEAP - Finite Element Analysis Program Published.* 2023. URL: http://projects.ce.berkeley.edu/feap/ (cit. on p. 113).

[221] Tekkaya, A. E., N. B. Khalifa, O. Hering, R. Meya, S. Myslicki, and F. Walther: ‘Forming-induced damage and its effects on product properties’. *CIRP Annals* (2017), vol. 66(1): pp. 281–284 (cit. on pp. 104, 105, 139–141, 218).

[222] Tortorelli, D. A. and P. Michaleris: ‘Design Sensitivity Analysis: Overview and Review’. *Inverse Problems in Engineering* (1994), vol. 1: pp. 71–103. DOI: 10.1080/174159794088027573 (cit. on p. 24).

[223] Tourki, Z. and K. Sai: 'Design and loading parameter optimisation in deep drawing process'. *International Journal of Vehicle Design* (2005), vol. 39(1-2): pp. 25–37. doi: 10.1504/IJVD.2005.007216 (cit. on p. 111).

[224] TRR188: *Transregio 188 "Damage controlled forming processes"*. 2023. url: https://trr188.de/index.php/en/ (cit. on pp. 3, 4, 102).

[225] Tvergaard, V. and A. Needleman: 'Analysis of the cup-cone fracture in a round tensile bar'. *Acta Metallurgica* (1984), vol. 32(1): pp. 157–169. doi: 10.1016/0001-6160(84)90213-X (cit. on p. 30).

[226] Van Rossum, G. and F. L. Drake Jr: *Python reference manual.* Centrum voor Wiskunde en Informatica Amsterdam, 1995 (cit. on p. 13).

[227] Venkadeshwaran, K., S. Das, and D. Misra: 'Bend angle prediction and parameter optimisation for laser bending of stainless steel using FEM and RSM'. *International Journal of Mechatronics and Manufacturing Systems* (2012), vol. 5(3-4): pp. 308–321. doi: 10.1504/IJMMS.2012.048237 (cit. on p. 110).

[228] Virtanen, P. et al.: 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. *Nature Methods* (2020), vol. 17: pp. 261–272. doi: 10.1038/s41592-019-0686-2 (cit. on pp. 13, 122).

[229] Wächter, A. and L. T. Biegler: 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming'. *Mathematical Programming* (2006), vol. 160: pp. 25–57. doi: 10.1007/s10107-004-0559-y (cit. on p. 20).

[230] Waffenschmidt, T., C. Polindara, A. Menzel, and S. Blanco: 'A gradient-enhanced large-deformation continuum damage model for fibre-reinforced materials'. *Computer Methods in Applied Mechanics and Engineering* (2014), vol. 268: pp. 801–842. doi: 10.1016/j.cma.2013.10.013 (cit. on pp. 30, 115).

[231] Wahed, M. A., A. K. Gupta, V. S. R. Gadi, S. K., S. K. Singh, and N. Kotkunde: 'Parameter optimisation in V-bending process at elevated temperatures to minimise spring back in Ti-6Al-4V alloy'. *Advances in Materials and Processing Technologies* (2020), vol. 6(2): pp. 350–364. doi: 10.1080/2374068X.2020.1728651 (cit. on p. 110).

[232] Waschinsky, N.: 'Structural optimisation of diffusion-driven degradation processes'. Dissertation. TU Dortmund University, 2021. doi: 10.17877/DE290R-22181 (cit. on p. 56).

[233] Wiechmann, K.: 'Theorie und Numerik zur Berechnung und Optimierung von Strukturen mit elastoplastischen Deformationen'. PhD thesis. Institut für Baumechanik und Numerische Mechanik, Universität Hannover, 2000 (cit. on pp. 4, 54, 55, 58, 60).

[234] Wiechmann, K. and E. Stein: ‘Shape optimization for elasto-plastic deformation under shakedown conditions’. *International Journal of Solids and Structures* (2006), vol. 43(22): pp. 7145–7165. doi: 10.1016/j.ijsolstr.2006.03.025 (cit. on p. 54).

[235] Wilson, E. A. and B. Parsons: ‘Finite element analysis of elastic contact problems using differential displacements’. *International Journal for Numerical Methods in Engineering* (1970), vol. 2(3): pp. 387–395. doi: 10.1002/nme.1620020307 (cit. on p. 101).

[236] Wohlgemuth, F. and F.-J. Barthold: ‘Controlling Physical Properties on Interfaces Using Parametrised Level Set Methods and Extended Finite Element Method’. *PAMM* (2019), vol. 19(1): e201900455. doi: 10.1002/pamm.201900455 (cit. on p. 4).

[237] Wriggers, P.: ‘Finite element algorithms for contact problems’. *Archives of Computational Methods in Engineering* (1995), vol. 2(4): pp. 1–49. doi: 10.1007/BF02736195 (cit. on p. 101).

[238] Wriggers, P.: *Nonlinear Finite Element Methods*. Springer Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-71001-1 (cit. on pp. 30, 33, 55).

[239] Yap, C. Y., C. K. Chua, Z. L. Dong, Z. H. Liu, D. Q. Zhang, L. E. Loh, and S. L. Sing: ‘Review of selective laser melting: Materials and applications’. *Applied Physics Reviews* (2015), vol. 2(4): p. 041101. doi: 10.1063/1.4935926 (cit. on p. 205).

[240] Ye, J., J. Becque, I. Hajirasouliha, S. M. Mojtabaei, and J. B. Lim: ‘Development of optimum cold-formed steel sections for maximum energy dissipation in uniaxial bending’. *Engineering Structures* (2018), vol. 161: pp. 55–67. doi: 10.1016/j.engstruct.2018.01.070 (cit. on p. 114).

[241] Yi, Y.-M., S.-H. Park, and S.-K. Youn: ‘Design of microstructures of viscoelastic composites for optimal damping characteristics’. *International Journal of Solids and Structures* (2000), vol. 37(35): pp. 4791–4810. doi: 10.1016/S0020-7683(99)00181-X (cit. on p. 4).

[242] Zubair, M., S. Sandlöbes-Haut, M. Wollenweber, K. Bugelnig, C. Kusche, G. Requena, and S. Korte-Kerzel: ‘Strain heterogeneity and micro-damage nucleation under tensile stresses in an Mg–5Al–3Ca alloy with an intermetallic skeleton’. *Materials Science and Engineering: A* (2019), vol. 767: p. 138414. doi: 10.1016/j.msea.2019.138414 (cit. on p. 3).

[243] Zuo, Z. H. and Y. M. Xie: ‘A simple and compact Python code for complex 3D topology optimization’. *Advances in Engineering Software* (2015), vol. 85: pp. 1–11. doi: 10.1016/j.advengsoft.2015.02.006 (cit. on p. 114).

List of Figures

List of Tables

Appendix A

On the implementation of the ductile damage model

In this appendix, the missing detailed equations regarding the ductile damage model in Chap. 4 are depicted. Furthermore, remarks on the utilisation of the $\bar{\boldsymbol{F}}$-method are given and supplemental results from computation with the implemented material model are compared to data from the original authors of [215].

A.1 Consistent material tangent operator

For the implementation of the ductile damage model, the consistent material tangent operators are required, see Fig. 4.4. Since the model is computed in principal space, see Sect. 4.6, the same notation regarding n-Tupels is utilised here as well. In addition to the introduced spatial identity tensor $\bar{\boldsymbol{I}}$, a diagonal matrix diag(1) of size $n \times n$ is introduced, where all diagonal values of are equal to one, with the remaining entries being equal to zero. Furthermore, the internal principal plastic strains $\bar{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}} - 2\Delta\lambda^{\mathrm{p}} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}$ are introduced here and utilised to compute the local quantities in principal space.

The total variations for the spatial Mandel stress and nonlocal damage driving forces resolve into two partial variations, i.e.

$$\delta \bar{\boldsymbol{m}} = \delta_{\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} \bar{\boldsymbol{m}} + \delta_{\phi} \bar{\boldsymbol{m}}, \tag{A.1}$$

$$\delta Y = \delta_{\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} Y + \delta_{\phi} Y. \tag{A.2}$$

where the partial variations can be identified as the tangent contributions and read

$$\delta_{\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} \bar{\boldsymbol{m}} = \left[\bar{\mathbb{C}}^{\mathrm{el}} + \bar{\mathbb{C}}^{\mathrm{pp}}\right] \cdot \delta \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}} = \frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} \cdot \delta \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}} + \left[\frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} + \frac{\partial \bar{\boldsymbol{m}}}{\partial d} \frac{\partial d}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}}\right] \cdot \delta \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}, \tag{A.3}$$

$$\delta_{\phi} \bar{\boldsymbol{m}} = \bar{\mathbb{C}}^{\mathrm{pd}} = \left[\frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \phi} + \frac{\partial \bar{\boldsymbol{m}}}{\partial d} \frac{\partial d}{\partial \phi}\right] \delta \phi, \tag{A.4}$$

$$\delta_{\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} Y = \bar{\mathbb{C}}^{\mathrm{dp}} = \frac{\partial Y}{\partial d} \frac{\partial d}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} \cdot \delta \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}, \tag{A.5}$$

$$\delta_{\phi} Y = \bar{\mathbb{C}}^{\mathrm{dd}} = \left[\beta_{\mathrm{d}} + \frac{\partial Y}{\partial d} \frac{\partial d}{\partial \phi}\right] \delta \phi. \tag{A.6}$$

The partial derivatives read

$$
\begin{aligned}
\frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} &= K f^{\mathrm{vol}}(d)\, \bar{\boldsymbol{I}} \otimes \bar{\boldsymbol{I}} + 2G f^{\mathrm{iso}}(d) \left[\mathrm{diag}(1) - \frac{1}{3} \bar{\boldsymbol{I}} \otimes \bar{\boldsymbol{I}}\right], && \text{(A.7)}\\
\frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} &= -2\, G f^{\mathrm{iso}}(d)\, \mathrm{diag}(1), && \text{(A.8)}\\
\frac{\partial \bar{\boldsymbol{m}}}{\partial d} &= K\, \left[f^{\mathrm{vol}}(d)\right]\, \bar{\boldsymbol{I}} + 2\, G \left[f^{\mathrm{iso}}(d)\right] \tilde{\varepsilon}\, \bar{\boldsymbol{\nu}}^{\mathrm{tr}}, && \text{(A.9)}\\
\frac{\partial Y}{\partial d} &= -\beta_{\mathrm{d}}, && \text{(A.10)}\\
\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} &= \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} \otimes \frac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} + \varDelta\lambda^{\mathrm{p}} \left[\hat{f}^{\boldsymbol{m}}\right]' \bar{\boldsymbol{\nu}}^{\mathrm{tr}} \otimes \frac{\mathrm{d}d}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} + \varDelta\lambda^{\mathrm{p}} \hat{f}^{\boldsymbol{m}} \frac{\partial \bar{\boldsymbol{\nu}}^{\mathrm{tr}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}}, && \text{(A.11)}\\
\frac{\partial d}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} &= \frac{\mathrm{d}\varDelta\lambda^{\mathrm{d}}}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} \hat{f}^{\alpha} + \varDelta\lambda^{\mathrm{d}} \left[\hat{f}^{\alpha}\right]' \sqrt{\frac{2}{3}} \frac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}}, && \text{(A.12)}\\
\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \phi} &= \frac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}\phi} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \varDelta\lambda^{\mathrm{p}} \left[\hat{f}^{\boldsymbol{m}}\right]' \frac{\partial d}{\partial \phi} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}, && \text{(A.13)}\\
\frac{\partial d}{\partial \phi} &= \frac{\mathrm{d}\varDelta\lambda^{\mathrm{d}}}{\mathrm{d}\phi} \hat{f}^{\alpha} + \varDelta\lambda^{\mathrm{d}} \left[\hat{f}^{\alpha}\right]' \sqrt{\frac{2}{3}} \frac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}\phi}, && \text{(A.14)}\\
\frac{\partial \bar{\boldsymbol{\nu}}^{\mathrm{tr}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} &= \|\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr,iso}}\|^{-1} \left[\mathrm{diag}(1) - \frac{1}{3} \bar{\boldsymbol{I}} \otimes \bar{\boldsymbol{I}} - \bar{\boldsymbol{\nu}}^{\mathrm{tr}} \otimes \bar{\boldsymbol{\nu}}^{\mathrm{tr}}\right]. && \text{(A.15)}
\end{aligned}
$$

where $\tilde{\varepsilon} = \|\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,iso}}\|$ is introduced as an abbreviation.

To compute the total derivatives of the Lagrangian, the implicit function theorem is required. Application at a point where the local Newton-Raphson is converged allows the specification of the required derivatives, such that

$$
\begin{aligned}
\frac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} &= -\det(\mathbf{J}^{-1}) \left[J_{\mathrm{pd}} \hat{f}^{\alpha} \frac{\partial \bar{\boldsymbol{m}}}{\partial d} + 2\, G J_{\mathrm{dd}}\, f^{\mathrm{iso}} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}\right], && \text{(A.16)}\\
\frac{\mathrm{d}\varDelta\lambda^{\mathrm{d}}}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} &= \quad \det(\mathbf{J}^{-1}) \left[J_{\mathrm{pp}} \hat{f}^{\alpha} \frac{\partial \bar{\boldsymbol{m}}}{\partial d} + 2\, G J_{\mathrm{dp}}\, f^{\mathrm{iso}} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}.\right], && \text{(A.17)}\\
\frac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}\phi} &= \quad \det(\mathbf{J}^{-1})\, J_{\mathrm{pd}} \beta_{\mathrm{d}} \hat{f}^{\alpha}, && \text{(A.18)}\\
\frac{\mathrm{d}\varDelta\lambda^{\mathrm{d}}}{\mathrm{d}\phi} &= -\det(\mathbf{J}^{-1})\, J_{\mathrm{pp}} \beta_{\mathrm{d}} \hat{f}^{\alpha}, && \text{(A.19)}
\end{aligned}
$$

where the specific Jacobi components are presented in the next section.

A.2 Additional data for local material point iteration

For the implementation of the ductile damage model on the local material point level, certain information have been omitted with in its respective section, see Sect. 4.5. To compute the Jacobian within the local Newton-Raphson scheme of the active-set algorithm, the Jacobian $\mathbf{J}$ is required, see Fig. 4.3. The specific contributions of (4.106) read

$$
\begin{aligned}
J_{\mathrm{pp}} &:= \left.\frac{\partial \mathit{\Phi}^{\mathrm{p}}}{\partial \mathit{\Delta}\lambda^{\mathrm{p}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} &&= \frac{\partial \mathit{\Phi}^{\mathrm{p}}}{\partial d}\,\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{p}}} - 2\,G\,f^{\mathrm{iso}}\left[\hat{f}^{\boldsymbol{m}}\right]^2 - \frac{2}{3}\frac{\partial \beta}{\partial \alpha}, && \text{(A.20)}\\
J_{\mathrm{pd}} &:= \left.\frac{\partial \mathit{\Phi}^{\mathrm{p}}}{\partial \mathit{\Delta}\lambda^{\mathrm{d}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} &&= \frac{\partial \mathit{\Phi}^{\mathrm{p}}}{\partial d}\,\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{d}}}, && \text{(A.21)}\\
J_{\mathrm{dp}} &:= \left.\frac{\partial \mathit{\Phi}^{\mathrm{d}}}{\partial \mathit{\Delta}\lambda^{\mathrm{p}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} &&= \left[2\,G\left[f^{\mathrm{iso}}\right]'\tilde{\varepsilon}\hat{f}^{\boldsymbol{m}} + \frac{\partial q}{\partial d}\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{p}}}\right]\hat{f}^{\alpha} + q\left[\hat{f}^{\alpha}\right]' \\
& && \quad + q_{\min}\left[f^{q}\right]' n_{\mathrm{d}}\left[1 - f^{q}\right]^{n_{\mathrm{d}}-1}\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{p}}}, && \text{(A.22)}\\
J_{\mathrm{d}} &:= \left.\frac{\partial \mathit{\Phi}^{\mathrm{d}}}{\partial \mathit{\Delta}\lambda^{\mathrm{d}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} &&= \frac{\partial q}{\partial d}\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{d}}}\hat{f}^{\alpha} + q_{\min}\left[f^{q}\right]' n_{\mathrm{d}}\left[1 - f^{q}\right]^{n_{\mathrm{d}}-1}\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{d}}}. && \text{(A.23)}
\end{aligned}
$$

with the additional partial derivatives

$$
\begin{aligned}
\frac{\partial \mathit{\Phi}^{\mathrm{p}}}{\partial d} &= 2\,G\tilde{\varepsilon}\left[\left[f^{\mathrm{iso}}\right]'\hat{f}^{\boldsymbol{m}} + f^{\mathrm{iso}}\left[\hat{f}^{\boldsymbol{m}}\right]'\right] - 2\,G\mathit{\Delta}\lambda^{\mathrm{p}} f^{\mathrm{iso}}\hat{f}^{\boldsymbol{m}}\left[\hat{f}^{\boldsymbol{m}}\right]', && \text{(A.24)}\\
\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{p}}} &= \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]', && \text{(A.25)}\\
\frac{\partial \beta}{\partial \alpha} &= h\,n_{\mathrm{p}}\alpha^{n_{\mathrm{p}}-1}, && \text{(A.26)}\\
\frac{\partial q}{\partial d} &= -\frac{K}{2}\left[f^{\mathrm{vol}}\right]''\left[\varepsilon^{\mathrm{e,vol}}\right]^2 - G\left[f^{\mathrm{iso}}\right]''\tilde{\varepsilon}^2 + \beta_{\mathrm{d}} + 2\,G\left[f^{\mathrm{iso}}\right]'\tilde{\varepsilon}\mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]', && \text{(A.27)}\\
\frac{\partial d}{\partial \mathit{\Delta}\lambda^{\mathrm{d}}} &= \hat{f}^{\alpha}. && \text{(A.28)}
\end{aligned}
$$

A.3 Remarks on the F-bar implementation

In [213] it was shown, that the used material model performs better regarding the nonlocal damage regularisation by implementing the $\bar{\boldsymbol{F}}$ method [71, 210]. This method is generally used to prevent volumetric locking in classic finite element (FE) environments to circumvent the (numerically expensive) use of higher order elements. For plasticity, it is used to prevent the overestimation in stiffness due to the deviatoric description of plasticity. Due to its definition, implementation of the $\bar{\boldsymbol{F}}$ method is relatively simple to incorporate in a standard FE environment. This method splits the deformation gradient $\boldsymbol{F}$ into a volumetric and isochoric part. The volumetric contribution in this method is exchanged with its counterpart evaluated at the element centroid, such that

$$
\bar{\boldsymbol{F}} = \left(\frac{J_0}{J}\right)^{\frac{1}{3}}\boldsymbol{F} \qquad \text{(A.29)}
$$

with

$$
J = \det(\boldsymbol{F}) \quad \text{and} \quad J_0 = \det(F_0). \qquad \text{(A.30)}
$$

Consequently, evaluation of local material point data with this input requiring leads to evaluation of tangent moduli data w.r.t. input $\bar{\boldsymbol{F}}$. Since the derivatives w.r.t. the original deformation gradient $\boldsymbol{F}$ are required for the FE implementation, the total variation of $\bar{\boldsymbol{F}}$ is required, i.e.

$$\begin{aligned}\delta\bar{\boldsymbol{F}} &= \frac{\partial\bar{\boldsymbol{F}}}{\partial\boldsymbol{F}} : \delta\boldsymbol{F} + \frac{\partial\bar{\boldsymbol{F}}}{\partial\boldsymbol{F}_0} : \delta\boldsymbol{F}_0 \\ &= \left(\frac{J_0}{J}\right)^{\frac{1}{3}} \frac{1}{3}\left[3\boldsymbol{I}\,\overline{\otimes}\,\boldsymbol{I} - \boldsymbol{F}\otimes\boldsymbol{F}^{-t}\right] : \delta\boldsymbol{F} + \left(\frac{J_0}{J}\right)^{\frac{1}{3}} \frac{1}{3}\left[\boldsymbol{F}\otimes{\boldsymbol{F}_0}^{-t}\right] : \delta\boldsymbol{F}_0 \\ &= \mathbb{C}_F : \delta\boldsymbol{F} + \mathbb{C}_{F_0} : \delta\boldsymbol{F}_0\end{aligned} \tag{A.31}$$

Applying this for example to (4.82), leads to

$$\begin{aligned}\boldsymbol{K}^{\varphi\varphi}_{eAC} = &\int_{\mathcal{B}^e_0} \left[\boldsymbol{I}\otimes\nabla_X N^{\varphi}_A\right] : \frac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}} \cdot \nabla_X N^{\varphi}_C \,\mathrm{d}V \\ &+ \int_{\mathcal{B}^e_0} \left[\boldsymbol{I}\otimes\nabla_X N^{\varphi}_A\right] : \frac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}_0} \cdot \nabla_X N^{\varphi}_{0,C} \,\mathrm{d}V,\end{aligned}$$

with

$$\frac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}} = \left.\frac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}}\right|_{\boldsymbol{F}=\bar{\boldsymbol{F}}} : \mathbb{C}_F, \tag{A.32}$$

$$\frac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}_0} = \left.\frac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}}\right|_{\boldsymbol{F}=\bar{\boldsymbol{F}}} : \mathbb{C}_{F_0}. \tag{A.33}$$

Consequently, this applies to all derivatives w.r.t. the deformation gradient, i.e. $\frac{\partial\bullet}{\partial\boldsymbol{F}}$, and is implicitly assumed, albeit not always directly mentioned. These also apply to the gradient information derived in Chap. 5 and has been shown to work to be applicable in the field of variational sensitivity analysis, cf. [99, 145, 146, 147]. The main benefit of this approach is, that it only requires small adjustments at the element level, which does not alter the constitutive equations at material point level.

A.4 Verification of the implemented model

To ensure the correct implementation of the presented ductile damage model within the used FE environment, the simulation data resulting from the implementation within this thesis is compared to the results from the original author [215]. To ensure the best accuracy, the same boundary value problem is taken from the original authors, see Fig. 6.8 in Sect. 6.3, and resulting data can be compared in a one-to-one fashion. The force-displacement curves are presented in Fig. A.1 and Fig. A.2. The results show almost perfect accordance, indicating correct implementation. The small deviations are a result of slight alterations in load application due to adjusted time step size.

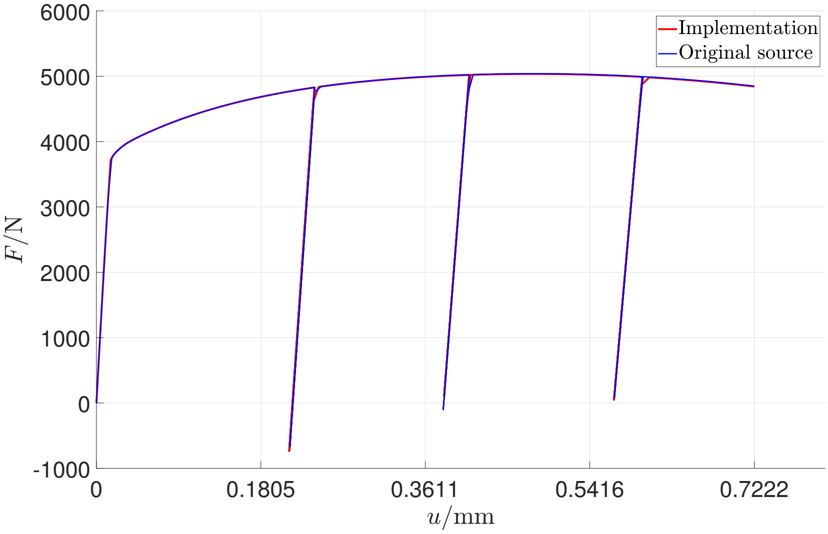


Figure A.1: Simulation data from [215] vs. results from the implemented model. The geometry is the notched plate.

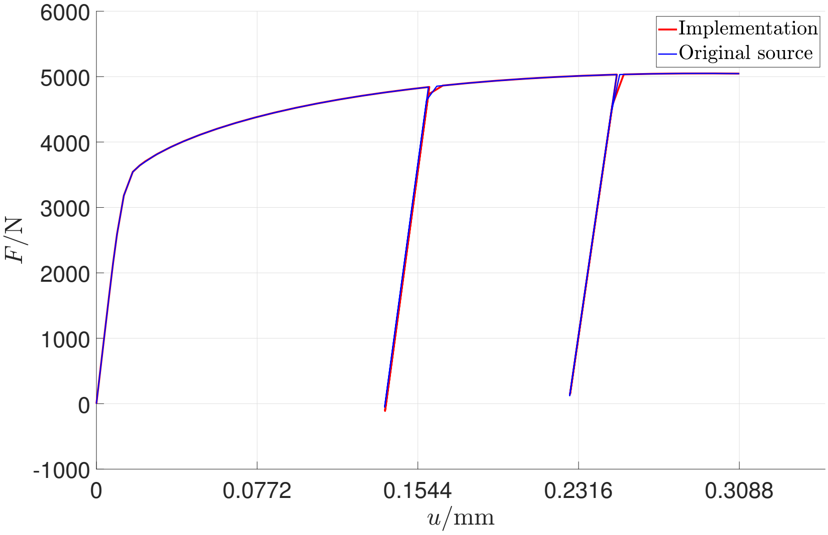


Figure A.2: Simulation data from [215] vs. results from the implemented model. The geometry is the plate with a hole.

Appendix B

Detailed derivation of sensitivity information

In this appendix, a detailed breakdown of the arising derivatives, which were omitted in Chap. 5, are derived in detail.

B.1 History sensitivities

In Chap. 5 the history sensitivities were derived. This term describes how the system reacts to a change in the history variables from the previous load step. In (5.17) and (5.18) the derivatives of the first Piola-Kirchhoff stress tensor $\boldsymbol{P}^{\mathrm{K}}$ and the scalar nonlocal damage driving force Y w.r.t. the history field $\boldsymbol{h}_n$ are required, which are derived in the following. The history field consists of the tensor valued quantity $\boldsymbol{C}^{\mathrm{p}\,-1}$, as well as the two scalar valued variables α and d. Since the model is computed in principal space, see Sect. 4.6, the same notation regarding n-Tuples is utilised here as well.

The first contribution is the partial variation w.r.t. the inverse plastic right Cauchy green tensor, i.e.

$$\delta_{\boldsymbol{C}_n^{\mathrm{p}-1}} \boldsymbol{P}^{\mathrm{K}} = \frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} : \delta \boldsymbol{C}_n^{\mathrm{p}-1}. \tag{B.1}$$

To compute the necessary derivative, using the definition of the spatial Mandel stress and relation (4.32), one can rewrite the above partial derivative, such that

$$\frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \left[\boldsymbol{I} \otimes \boldsymbol{F}^{-1}\right] : \left[\frac{\partial \bar{\boldsymbol{m}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}\right]. \tag{B.2}$$

Since the Mandel stress $\bar{\boldsymbol{m}}$ are defined and calculated in the principal space, the above unknown derivative is split into two components using the chain rule

$$\frac{\partial \bar{\boldsymbol{m}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \frac{\partial \bar{\boldsymbol{m}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}, \tag{B.3}$$

where the latter derivative can be specified as

$$\frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = [\boldsymbol{I} \overline{\otimes} \boldsymbol{F}] : [\boldsymbol{F} \cdot \mathbb{I}^{\mathrm{sym}}], \tag{B.4}$$

with the symmetric 4th-order identity tensor

$$\mathbb{I}^{\text{sym}} = \frac{1}{2}\left[\boldsymbol{I}\overline{\otimes}\boldsymbol{I} + \boldsymbol{I}\underline{\otimes}\boldsymbol{I}\right]. \tag{B.5}$$

The latter derivative in (B.3) splits into the derivative of the eigenvalues and eigenvectors

$$\frac{\partial \bar{\boldsymbol{m}}}{\partial \boldsymbol{b}^{\text{e,tr}}} = \sum_{i=1}^{3}\sum_{j=1}^{3}\frac{\partial m_i}{\partial b_j^{\text{e,tr}}}\boldsymbol{P}_i \otimes \boldsymbol{P}_j + \sum_{i=1}^{3} m_i \frac{\partial \boldsymbol{P}_i}{\partial \boldsymbol{b}^{\text{e,tr}}}, \tag{B.6}$$

where the eigenprojections are introduced as

$$\boldsymbol{P}_i = \boldsymbol{n}_i \otimes \boldsymbol{n}_i. \tag{B.7}$$

The first, scalar valued derivative can be computed using (A.7), such that

$$\frac{\partial m_i}{\partial b_j^{\text{e,tr}}} = \frac{\partial m_i}{\partial \varepsilon_i^{\text{e,tr}}}\frac{1}{2\left[\lambda_j^{\text{e,tr}}\right]^2}. \tag{B.8}$$

The remaining derivatives of the eigenprojections are purely kinematic and can be derived by combination of Sylvester's formula and the Cayley-Hamilton theorem, cf. [117], which results for an arbitrary tensor $\boldsymbol{T}$ with corresponding eigenvalues λ in

$$\begin{aligned}\frac{\partial \boldsymbol{P}_i}{\partial \boldsymbol{T}} = \frac{\lambda_i}{D_i}\Bigg[&\mathbb{I}^{\text{sym}} - \boldsymbol{I}\otimes\boldsymbol{I} + \frac{1}{\lambda_i}J_3^{\boldsymbol{T}}\left[\mathbb{I}^{\text{sym}}_{\boldsymbol{T}^{-1}} + \boldsymbol{T}^{-1}\otimes\boldsymbol{T}^{-1}\right] + \boldsymbol{I}\otimes\boldsymbol{P}_i + \boldsymbol{P}_i\otimes\boldsymbol{I} \\ &- \frac{1}{\lambda_i^2}J_3^{\boldsymbol{T}}\left[\boldsymbol{T}^{-1}\otimes\boldsymbol{P}_i + \boldsymbol{P}_i\otimes\boldsymbol{T}^{-1}\right] + \frac{1}{\lambda_i}\left[\frac{1}{\lambda_i}D_i - D_i'\right]\boldsymbol{P}_i\otimes\boldsymbol{P}_i\Bigg],\end{aligned} \tag{B.9}$$

with the helper terms

$$D_i = 2\,\lambda_i^2 - \lambda_i\,J_1^{\boldsymbol{T}} + \frac{1}{\lambda_i}J_3^{\boldsymbol{T}}, \tag{B.10}$$

$$D_i' = 4\,\lambda_i - J_1^{\boldsymbol{T}} - \frac{1}{\lambda_i^2}J_3^{\boldsymbol{T}}, \tag{B.11}$$

$$J_1^{\boldsymbol{T}} = \lambda_1 + \lambda_2 + \lambda_3 = \operatorname{tr}(\boldsymbol{T}), \tag{B.12}$$

$$J_3^{\boldsymbol{T}} = \lambda_1\,\lambda_2\,\lambda_3 = \det(\boldsymbol{T}), \tag{B.13}$$

$$\mathbb{I}^{\text{sym}}_{\boldsymbol{T}^{-1}} = -\frac{1}{2}\left[\boldsymbol{T}^{-1}\overline{\otimes}\boldsymbol{T}^{-1} + \boldsymbol{T}^{-1}\underline{\otimes}\boldsymbol{T}^{-1}\right]. \tag{B.14}$$

For the underlying problem in (B.6), $\boldsymbol{T} = \boldsymbol{b}^{\text{e,tr}}$ and $\lambda = [\lambda^{\text{e,tr}}]^2$ have to be inserted above. It has to be noted, that this only holds for three distinctly different eigenvalues $\lambda_i^{\text{e,tr}}$. In the implementation this is enforced by numerically perturbing the eigenvalues, if their values are too similar. In turn, under certain conditions of the boundary value problem, this might lead to slight errors in the resulting derivatives.

Following the same approach for the nonlocal scalar valued damage driving force Y, incorporating (A.5), one gets

$$\frac{\partial Y}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \left[\sum_{i=1}^{3} \frac{\partial Y}{\partial \varepsilon_i^{\mathrm{e,tr}}} \frac{1}{2 \left[\lambda_j^{\mathrm{e,tr}}\right]^2} \boldsymbol{P}_i \right] : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}. \tag{B.15}$$

Similar to (B.2), the partial variation of $\boldsymbol{P}^{\mathrm{K}}$ w.r.t α_n reduces to the derivative of the Mandel stresses, hence

$$\frac{\partial \boldsymbol{P}^{\mathrm{K}}}{\partial \alpha_n} = \frac{\partial \bar{\boldsymbol{m}}}{\partial \alpha_n} \cdot \boldsymbol{F}^{-t}. \tag{B.16}$$

The first derivative has to be further split, and computation in principal space yields

$$\frac{\partial \bar{\boldsymbol{m}}}{\partial \alpha_n} = \frac{\partial \bar{\boldsymbol{m}}}{\partial d} \frac{\partial d}{\partial \alpha_n} + \frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \alpha_n}, \tag{B.17}$$

with the partial derivatives from (A.8) and (A.9). The partial derivatives of local damage and plastic strains result in

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \alpha_n} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\alpha_n} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}} \left[\hat{f}^{\boldsymbol{m}}\right]' \frac{\partial d}{\partial \alpha_n} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}, \tag{B.18}$$

$$\frac{\partial d}{\partial \alpha_n} = \Delta\lambda^{\mathrm{d}} \left[\hat{f}^{\alpha}\right]' \frac{\partial \alpha}{\partial \alpha_n} + \hat{f}^{\alpha} \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}\alpha_n}, \tag{B.19}$$

$$\frac{\partial \alpha}{\partial \alpha_n} = 1 + \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\alpha_n}. \tag{B.20}$$

To calculate the derivatives of the Lagrange multipliers w.r.t. α_n, one has to solve the following system of equations by using the same implicit function theorem applied to the converged local Newton-Raphson in Appendix A.2 and its Jacobian $\mathbf{J}$, i.e.

$$\begin{bmatrix} \dfrac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\alpha_n} \\ \dfrac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}\alpha_n} \end{bmatrix} = -\,\mathbf{J}^{-1} \cdot \left. \begin{bmatrix} \dfrac{\partial \Phi^{\mathrm{p}}}{\partial \alpha_n} \\ \dfrac{\partial \Phi^{\mathrm{d}}}{\partial \alpha_n} \end{bmatrix} \right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} . \tag{B.21}$$

The required derivatives of the dissipation potentials result in

$$\left.\frac{\partial \Phi^{\mathrm{p}}}{\partial \alpha_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2\,G\left[\tilde{\varepsilon}\left[\left[f^{\mathrm{iso}}\right]'\hat{f}^{\boldsymbol{m}} + f^{\mathrm{iso}}\left[\hat{f}^{\boldsymbol{m}}\right]'\right]\right.$$
$$\left. - f^{\mathrm{iso}}\hat{f}^{\boldsymbol{m}}\Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\right]\frac{\partial d}{\partial \alpha_n} - \sqrt{\frac{2}{3}}\frac{\partial \beta}{\partial \alpha_n}, \tag{B.22}$$

$$\left.\frac{\partial d}{\partial \alpha_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \Delta\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]', \tag{B.23}$$

$$\left.\frac{\partial \beta}{\partial \alpha_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = h\,n_{\mathrm{p}}\,\alpha^{n_{\mathrm{p}}-1}, \tag{B.24}$$

and

$$\left.\frac{\partial \Phi^{\mathrm{d}}}{\partial \alpha_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial \alpha_n}\hat{f}^{\alpha} + q\left[\hat{f}^{\alpha}\right]' + q_{\min}\left[f^{q}\right]' n_{\mathrm{d}}\left[1 - f^{q}\right]^{n_{\mathrm{d}}-1}\frac{\partial d}{\partial \alpha_n}, \tag{B.25}$$

$$\left.\frac{\partial q}{\partial \alpha_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \left[\frac{\partial q}{\partial d} + 2\,G\left[f^{\mathrm{iso}}\right]'\tilde{\varepsilon}\Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\right]\frac{\partial d}{\partial \alpha_n}, \tag{B.26}$$

$$\left.\frac{\partial q}{\partial d}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -\frac{K}{2}\left[f^{\mathrm{vol}}\right]''\left[\varepsilon^{\mathrm{e,vol}}\right]^2 - G\left[f^{\mathrm{iso}}\right]''\tilde{\varepsilon}^2 - \beta_{\mathrm{d}}, \tag{B.27}$$

using the partial derivative (A.27). The derivative of the scalar valued nonlocal damage driving force reduces to

$$\frac{\partial Y}{\partial \alpha_n} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial \alpha_n}, \tag{B.28}$$

using (B.19).

The same line of deduction, i.e. (B.16), is applied for the derivatives w.r.t. d_n, i.e.

$$\frac{\partial \boldsymbol{m}}{\partial d_n} = \frac{\partial \bar{\boldsymbol{m}}}{\partial d}\frac{\partial d}{\partial d_n} + \frac{\partial \bar{\boldsymbol{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial d_n}, \tag{B.29}$$

with

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial d_n} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}d_n}\hat{f}^{\boldsymbol{m}}\bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial d_n}\bar{\boldsymbol{\nu}}^{\mathrm{tr}}, \tag{B.30}$$

$$\frac{\partial d}{\partial d_n} = 1 + \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}d_n}\hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\frac{\partial \alpha}{\partial d_n}, \tag{B.31}$$

$$\frac{\partial \alpha}{\partial d_n} = \sqrt{\frac{2}{3}}\frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}d_n}. \tag{B.32}$$

Again, to solve for the derivatives of the Lagrange multipliers, the system of equation

$$\begin{bmatrix} \dfrac{\mathrm{d}\varDelta\lambda^{\mathrm{p}}}{\mathrm{d}d_n} \\ \dfrac{\mathrm{d}\varDelta\lambda^{\mathrm{d}}}{\mathrm{d}d_n} \end{bmatrix} = -\,\mathbf{J}^{-1}\cdot \left.\begin{bmatrix} \dfrac{\partial\varPhi^{\mathrm{p}}}{\partial d_n} \\ \dfrac{\partial\varPhi^{\mathrm{d}}}{\partial d_n} \end{bmatrix}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}}\qquad, \tag{B.33}$$

has to be solved. The derivatives of the dissipation potentials herein read

$$\left.\frac{\partial\varPhi^{\mathrm{p}}}{\partial d_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2\,G\tilde{\varepsilon}\left[\left[f^{\mathrm{iso}}\right]'\hat{f}^{\boldsymbol{m}} + f^{\mathrm{iso}}\left[\hat{f}^{\boldsymbol{m}}\right]'\right] - 2\,Gf^{\mathrm{iso}}\hat{f}^{\boldsymbol{m}}\varDelta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]', \tag{B.34}$$

$$\left.\frac{\partial\varPhi^{\mathrm{d}}}{\partial d_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial d_n}\hat{f}^{\alpha} + q_{\mathrm{min}}\left[f^{q}\right]' n_{\mathrm{d}}\left[1-f^{q}\right]^{n_{\mathrm{d}}-1}, \tag{B.35}$$

$$\left.\frac{\partial q}{\partial d_n}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial d} + 2\,G\left[f^{\mathrm{iso}}\right]'\tilde{\varepsilon}\varDelta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'. \tag{B.36}$$

And applying this method to the derivative of Y, this leads to

$$\frac{\partial Y}{\partial d_n} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial d_n}, \tag{B.37}$$

by using (B.31).

B.2 Geometric history sensitivity update

After computation of the sensitivity matrix $\mathbf{S}$, an update for the history variations is required. For the derivation of this update in (5.19), the derivatives of the three history variables $\boldsymbol{h} = \left\{\boldsymbol{C}^{\mathrm{p}^{-1}},\alpha,d\right\}$ are required. They are derived and specified in the following subsections.

B.2.1 Derivatives with respect to the deformation gradient

In order to compute the derivatives w.r.t. the deformation gradient $\boldsymbol{F}$, the chain rule is applied, since only $\boldsymbol{b}^{\mathrm{e,tr}}$ is explicitly dependant on the deformation gradient. Therefore, any derivative can be written as

$$\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{F}} = \frac{\partial\boldsymbol{h}}{\partial\boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial\boldsymbol{b}^{\mathrm{e,tr}}}{\partial\boldsymbol{F}}, \tag{B.38}$$

with the latter resulting in

$$\frac{\partial\boldsymbol{b}^{\mathrm{e,tr}}}{\partial\boldsymbol{F}} = \boldsymbol{I}\,\overline{\otimes}\,\boldsymbol{F}\cdot\boldsymbol{C}_n^{\mathrm{p}^{-1}} + \boldsymbol{F}\cdot\boldsymbol{C}_n^{\mathrm{p}^{-1}}\,\underline{\otimes}\,\boldsymbol{I}, \tag{B.39}$$

while the first derivative has to be specified for each respective history variable $\boldsymbol{h}$.

Starting with the inverse plastic right Cauchy-Green deformation tensor, one obtains

$$\frac{\partial \boldsymbol{C}^{\mathrm{p}-1}}{\partial \boldsymbol{F}} = \left[\boldsymbol{I} \,\overline{\otimes}\, \left[\boldsymbol{b}^{\mathrm{e}} \cdot \boldsymbol{F}^{-t}\right]^{t}\right] : \frac{\partial \boldsymbol{F}^{-1}}{\partial \boldsymbol{F}} + \boldsymbol{F}^{-1} \cdot \boldsymbol{b}^{\mathrm{e}} \cdot \frac{\partial \boldsymbol{F}^{-t}}{\partial \boldsymbol{F}} + \left[\boldsymbol{F}^{-1} \,\overline{\otimes}\, \boldsymbol{F}^{-1}\right] : \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{F}}, \tag{B.40}$$

where the derivatives of the inverse, and inverse transposed deformation gradient read

$$\frac{\partial \boldsymbol{F}^{-1}}{\partial \boldsymbol{F}} = -\boldsymbol{F}^{-1} \,\overline{\otimes}\, \boldsymbol{F}^{-t}, \tag{B.41}$$

$$\frac{\partial \boldsymbol{F}^{-t}}{\partial \boldsymbol{F}} = -\boldsymbol{F}^{-t} \,\underline{\otimes}\, \boldsymbol{F}^{-1}, \tag{B.42}$$

respectively. For the derivative of the elastic left Cauchy-Green tensor, application of the chain rule again yields a known deformation gradient dependent quantity, and a quantity dependant of the local model, i.e.

$$\frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{F}} = \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{F}}. \tag{B.43}$$

Application of the concept described in (B.6) regarding eigenvalues and -projections, leads to

$$\frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} = \sum_{i=1}^{3}\sum_{j=1}^{3} \exp(2\,\varepsilon_i^{\mathrm{e}}) \frac{\partial \varepsilon_i^{\mathrm{e}}}{\partial \varepsilon_j^{\mathrm{e,tr}}} \frac{1}{\left[\lambda_j^{\mathrm{e,tr}}\right]^2} \boldsymbol{P}_i \otimes \boldsymbol{P}_j + \sum_{i=1}^{3} \exp(2\,\varepsilon_i^{\mathrm{e}}) \frac{\partial \boldsymbol{P}_i}{\partial \boldsymbol{b}^{\mathrm{e,tr}}}, \tag{B.44}$$

with

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} = \boldsymbol{I} - \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}}, \tag{B.45}$$

and the additional derivative already defined (A.11).

The derivatives of α and d require their respective evolution equations which in turn leads to the derivative of the Lagrangian multiplier w.r.t. the trial elastic strains, i.e.

$$\frac{\partial \alpha}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} = \sum_{i=1}^{3} \frac{\partial \alpha}{\partial \varepsilon_i^{\mathrm{e,tr}}} \frac{1}{2\left[\lambda_i^{\mathrm{e,tr}}\right]^2} \boldsymbol{P}_i, \tag{B.46}$$

$$\frac{\partial d}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} = \sum_{i=1}^{3} \frac{\partial d}{\partial \varepsilon_i^{\mathrm{e,tr}}} \frac{1}{2\left[\lambda_i^{\mathrm{e,tr}}\right]^2} \boldsymbol{P}_i, \tag{B.47}$$

$$\frac{\partial \alpha}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}} = \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\bar{\boldsymbol{\varepsilon}}^{\mathrm{e,tr}}}, \tag{B.48}$$

utilising (A.12) and (A.16).

Remark B.1 The line of deduction used in (B.38) can in this sense also be applied to the material tangent $\partial \boldsymbol{P}^{\mathrm{K}}/\partial \boldsymbol{F}$ within the stiffness contribution $\mathbf{K}_e^{\varphi\varphi}$, see (4.82). Since another derivation is utilised in the original paper to calculate this tangent term, it is therefore possible to compare both results and verify their correct implementation.

B.2.2 Derivatives with respect to nonlocal damage

Following the same approach for the derivatives w.r.t. the nonlocal damage variable ϕ, the derivative for $\boldsymbol{C}^{\mathrm{p}-1}$ reduces to

$$\frac{\partial \boldsymbol{C}^{\mathrm{p}-1}}{\partial \phi} = \boldsymbol{F}^{-1} \cdot \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \phi} \cdot \boldsymbol{F}^{-t}, \tag{B.49}$$

$$\frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \phi} = \sum_{i=1}^{3} 2\exp(2\varepsilon_i^{\mathrm{e}}) \frac{\partial \varepsilon_i^{\mathrm{p}}}{\partial \phi} \boldsymbol{P}_i, \tag{B.50}$$

which requires the derivate of the local principal plastic strains (A.13). The derivative of local damage w.r.t. nonlocal damage is required for the tangent contributions and therefore is already defined in (A.14). The derivative of α finally reads

$$\frac{\partial \alpha}{\partial \phi} = \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\phi}, \tag{B.51}$$

with the derivative of the Lagrangian multiplier (A.18).

B.2.3 Derivatives with respect to the history field

The computation of the derivatives w.r.t the history field from the previous load step n follows in the same steps as the previous derivations and thus utilises many already established quantities. For the derivatives w.r.t. $\boldsymbol{C}_n^{\mathrm{p}-1}$, starting with $\boldsymbol{C}^{\mathrm{p}-1}$, this leads to

$$\frac{\partial \boldsymbol{C}^{\mathrm{p}-1}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \left[\boldsymbol{F}^{-1} \overline{\otimes} \boldsymbol{F}^{-1}\right] : \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}. \tag{B.52}$$

The unknown derivative can be easily computed by using the chain rule

$$\frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}, \tag{B.53}$$

which only consists of known quantities, i.e. (B.4) and (B.44).

The same applies to the internal hardening variable α and the local damage d, i.e.

$$\frac{\partial \alpha}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \frac{\partial \alpha}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}, \tag{B.54}$$

$$\frac{\partial d}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \frac{\partial d}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}, \tag{B.55}$$

which both combine (B.4) with (B.46) and (B.47), respectively.

The derivatives w.r.t. α_n read

$$\frac{\partial \boldsymbol{C}^{\mathrm{p}\,-1}}{\partial \alpha_n} = \boldsymbol{F}^{-1} \cdot \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \alpha_n} \cdot \boldsymbol{F}^{-t}, \tag{B.56}$$

with

$$\frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial \alpha_n} = \sum_{i=1}^{3} 2\exp(2\varepsilon_i^{\mathrm{e}}) \frac{\partial \varepsilon_i^{\mathrm{e}}}{\partial \alpha_n} \boldsymbol{P}_i, \tag{B.57}$$

$$\frac{\partial \bar{\varepsilon}^{\mathrm{e}}}{\partial \alpha_n} = -\frac{\partial \bar{\varepsilon}^{\mathrm{p}}}{\partial \alpha_n}, \tag{B.58}$$

where the latter derivative has been specified in (B.18). The remaining two derivatives w.r.t. α_n, i.e. $\partial \alpha / \partial \alpha_n$ and $\partial d / \partial \alpha_n$, can be found in (B.20) and (B.19), respectively.

Finally, the derivatives w.r.t. the damage variable from the previous load step result in

$$\frac{\partial \boldsymbol{C}^{\mathrm{p}\,-1}}{\partial d_n} = \boldsymbol{F}^{-1} \cdot \frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial d_n} \cdot \boldsymbol{F}^{-t}, \tag{B.59}$$

with

$$\frac{\partial \boldsymbol{b}^{\mathrm{e}}}{\partial d_n} = \sum_{i=1}^{3} 2\exp(2\varepsilon_i^{\mathrm{e}}) \frac{\partial \varepsilon_i^{\mathrm{e}}}{\partial d_n} \boldsymbol{P}_i, \tag{B.60}$$

$$\frac{\partial \bar{\varepsilon}^{\mathrm{e}}}{\partial d_n} = -\frac{\partial \bar{\varepsilon}^{\mathrm{p}}}{\partial d_n}, \tag{B.61}$$

which is further specified in (B.30). The remaining two derivatives for the history update, i.e. $\mathrm{d}\alpha / \partial d_n$ and $\mathrm{d}d / \partial d_n$, can be found in (B.32) and (B.31), respectively.

B.3 Material sensitivity data

In Chap. 5, the derivatives of the nonlocal driving forces w.r.t. the constitutive parameters $\mathbf{m}_{\mathrm{p}}$ are required. Due to the excessive number of material parameters within the considered model, it was not feasible to list all the resulting derivatives in the specific section. Hence, they are derived here. They are listed in contributing order, i.e. elastic, plastic, damage and regularising, respectively. Only the pseudo load contributions are derived here, as the derivation of the history derivatives is straightforward and can be found in its respective section. The quantities here are listed for sake of completeness and no specific deduction of applied methodology is made here, as it follows the same concepts as in the previous section. Therefore, the relation $\partial \mathbf{P}^{\mathrm{K}} / \partial \bullet = \partial \bar{\mathbf{m}} / \partial \bullet \cdot \mathbf{F}^{-t}$ has to considered as it is not explicitly stated again in every following subsection. In contrast to the other derivatives in this appendix, the following quantities in this section are explicitly discrete ones. Since the material parameters are inherently discrete, the partial derivatives can only be computed in a discrete manner as well.

B.3.1 Elastic parameters

As done prior, the starting point is again the correlation between the spatial Mandel stress and the first Piola-Kichhoff stress tensor, which in return requires the derivatives of the Mandel stress w.r.t. the elastic material parameters, which read

$$\frac{\partial \bar{\mathbf{m}}}{\partial E} = \frac{\partial K}{\partial E} f^{\mathrm{vol}} \varepsilon^{\mathrm{e,vol}} \bar{\mathbf{I}} + 2 \frac{\partial G}{\partial E} f^{\mathrm{iso}} \tilde{\varepsilon} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \frac{\partial \bar{\mathbf{m}}}{\partial d} \frac{\partial d}{\partial E} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial E} \tag{B.62}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial \nu} = \frac{\partial K}{\partial \nu} f^{\mathrm{vol}} \varepsilon^{\mathrm{e,vol}} \bar{\mathbf{I}} + 2 \frac{\partial G}{\partial \nu} f^{\mathrm{iso}} \tilde{\varepsilon} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \frac{\partial \bar{\mathbf{m}}}{\partial d} \frac{\partial d}{\partial \nu} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \nu}, \tag{B.63}$$

wherein already derived partial derivatives (A.8) and(A.9) are used. The additional partial derivatives read

$$\frac{\partial K}{\partial E} = \frac{1}{3 - 6\nu} \quad \text{and} \quad \frac{\partial G}{\partial E} = \frac{1}{2\,[1+\nu]}, \tag{B.64}$$

$$\frac{\partial K}{\partial \nu} = \frac{2\,E}{3(2-2\nu)^2} \quad \text{and} \quad \frac{\partial G}{\partial \nu} = -\frac{E}{2\,[1+\nu]^2}, \tag{B.65}$$

assuming $0 < \nu < 0.5$. The remaining derivatives yield

$$\frac{\partial d}{\partial E} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}E} \hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}} \left[\hat{f}^{\alpha}\right]' \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}E}, \tag{B.66}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial E} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}E} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}} \left[\hat{f}^{\boldsymbol{m}}\right]' \frac{\partial d}{\partial E} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}, \tag{B.67}$$

$$\frac{\partial d}{\partial \nu} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}\nu} \hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}} \left[\hat{f}^{\alpha}\right]' \sqrt{\frac{2}{3}} \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\nu}, \tag{B.68}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \nu} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\nu} \hat{f}^{\boldsymbol{m}} \bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}} \left[\hat{f}^{\boldsymbol{m}}\right]' \frac{\partial d}{\partial \nu} \bar{\boldsymbol{\nu}}^{\mathrm{tr}}, \tag{B.69}$$

The additional derivatives require the Lagrangian multipliers derivatives and hence application of the implicit function theorem for the local Newton-Raphson, such that

$$\begin{bmatrix} \dfrac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\bullet} \\ \dfrac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}\bullet} \end{bmatrix} = -\,\mathbf{J}^{-1} \cdot \left. \begin{bmatrix} \dfrac{\partial \Phi^{\mathrm{p}}}{\partial \bullet} \\ \dfrac{\partial \Phi^{\mathrm{d}}}{\partial \bullet} \end{bmatrix} \right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} . \tag{B.70}$$

The derivatives of the dissipation potentials read

$$\left. \frac{\partial \Phi^{\mathrm{p}}}{\partial E} \right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2 \frac{\partial G}{\partial E} f^{\mathrm{iso}} \hat{f}^{\boldsymbol{m}} \tilde{\varepsilon} \quad \text{and} \quad \left. \frac{\partial \Phi^{\mathrm{d}}}{\partial E} \right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial E} \hat{f}^{\alpha}, \tag{B.71}$$

$$\left. \frac{\partial \Phi^{\mathrm{p}}}{\partial \nu} \right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2 \frac{\partial G}{\partial \nu} f^{\mathrm{iso}} \hat{f}^{\boldsymbol{m}} \tilde{\varepsilon} \quad \text{and} \quad \left. \frac{\partial \Phi^{\mathrm{d}}}{\partial \nu} \right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial \nu} \hat{f}^{\alpha}. \tag{B.72}$$

with the derivatives

$$\frac{\partial q}{\partial E} = -\frac{1}{2}\frac{\partial K}{\partial E}\left[f^{\mathrm{vol}}\right]'\left[\varepsilon^{\mathrm{e,vol}}\right]^2 - \frac{\partial G}{\partial E} f^{\mathrm{iso}}\tilde{\varepsilon}^2, \tag{B.73}$$

$$\frac{\partial q}{\partial \nu} = -\frac{1}{2}\frac{\partial K}{\partial \nu}\left[f^{\mathrm{vol}}\right]'\left[\varepsilon^{\mathrm{e,vol}}\right]^2 - \frac{\partial G}{\partial \nu} f^{\mathrm{iso}}\tilde{\varepsilon}^2. \tag{B.74}$$

The derivatives considering the scalar nonlocal driving in return yield

$$\frac{\partial Y}{\partial E} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial E} \quad \text{and} \quad \frac{\partial Y}{\partial \nu} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial \nu}. \tag{B.75}$$

B.3.2 Plastic parameters

This section considers the derivatives of the three plastic parameters σ_{y0}, h and n_{p}, i.e. the yield stress, the hardening modulus and the hardening exponent, respectively. The derivative of the spatial Mandel stress reads

$$\frac{\partial \bar{\mathbf{m}}}{\partial \sigma_{\mathrm{y0}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \sigma_{\mathrm{y0}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \sigma_{\mathrm{y0}}}, \tag{B.76}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial h} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial h} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial h}, \tag{B.77}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial n_{\mathrm{p}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial n_{\mathrm{p}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial n_{\mathrm{p}}}, \tag{B.78}$$

with the additional derivatives

$$\frac{\partial d}{\partial \sigma_{\mathrm{y0}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}\sigma_{\mathrm{y0}}}\hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\sigma_{\mathrm{y0}}}, \tag{B.79}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \sigma_{\mathrm{y0}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\sigma_{\mathrm{y0}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \sigma_{\mathrm{y0}}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.80}$$

$$\frac{\partial d}{\partial h} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}h}\hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}h}, \tag{B.81}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial h} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}h}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial h}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.82}$$

$$\frac{\partial d}{\partial n_{\mathrm{p}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}n_{\mathrm{p}}}\hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}n_{\mathrm{p}}}, \tag{B.83}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial n_{\mathrm{p}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}n_{\mathrm{p}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial n_{\mathrm{p}}}\bar{\mathbf{v}}^{\mathrm{tr}}. \tag{B.84}$$

Again, the implicit function theorem is used, wherein the dissipation potentials read

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \sigma_{\mathrm{y0}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -\sqrt{\frac{2}{3}} \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \sigma_{\mathrm{y0}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0, \tag{B.85}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial h}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -\sqrt{\frac{2}{3}}\alpha^{n_{\mathrm{p}}} \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial h}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0, \tag{B.86}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial n_{\mathrm{p}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -\sqrt{\frac{2}{3}}\ln(\alpha)\beta \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial n_{\mathrm{p}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0. \tag{B.87}$$

The derivatives considering the scalar nonlocal driving in return yields

$$\frac{\partial Y}{\partial \sigma_{\mathrm{y0}}} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial \sigma_{\mathrm{y0}}}, \quad \frac{\partial Y}{\partial h} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial h}, \quad \frac{\partial Y}{\partial n_{\mathrm{p}}} = -\beta_{\mathrm{d}}\frac{\partial d}{\partial n_{\mathrm{p}}} \tag{B.88}$$

B.3.3 Damage parameters

The damage parameters make up more than half of the constitutive parameters. Therefore, this section will be quite lengthy. The considered parameters for the derivatives here are the damage rate factor η, the volumetric ξ_{vol} and isochoric ξ_{iso} damage factors, the threshold factor ξ_{q}, the effective stress factor ξ_{m}, the coupling factor η_α, the damage threshold q_{min} and the damage exponent n_{d}. The derivatives of the spatial Mandel stresses read

$$\frac{\partial \bar{\mathbf{m}}}{\partial \eta} = -K\xi_{\mathrm{vol}}\, d\, f^{\mathrm{vol}}\varepsilon^{\mathrm{e,vol}}\bar{\mathbf{I}} - 2\,G\xi_{\mathrm{iso}}\, d\, f^{\mathrm{iso}}\tilde{\varepsilon}\bar{\mathbf{v}}^{\mathrm{tr}} + \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \eta} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \eta}, \tag{B.89}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial \xi_{\mathrm{vol}}} = -K\eta\, d\, f^{\mathrm{vol}}\varepsilon^{\mathrm{e,vol}}\bar{\mathbf{I}} + \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \xi_{\mathrm{vol}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{vol}}}, \tag{B.90}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial \xi_{\mathrm{iso}}} = -2\,G\eta\, d\, f^{\mathrm{iso}}\tilde{\varepsilon}\bar{\mathbf{v}}^{\mathrm{tr}} + \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \xi_{\mathrm{iso}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{iso}}}, \tag{B.91}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial \xi_{\mathrm{q}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \xi_{\mathrm{q}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{q}}}, \tag{B.92}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial \xi_{\mathrm{m}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \xi_{\mathrm{m}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{m}}}, \tag{B.93}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial \eta_\alpha} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial \eta_\alpha} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \eta_\alpha}, \tag{B.94}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial q_{\mathrm{min}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial q_{\mathrm{min}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial q_{\mathrm{min}}}, \tag{B.95}$$

$$\frac{\partial \bar{\mathbf{m}}}{\partial n_{\mathrm{d}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d}\frac{\partial d}{\partial n_{\mathrm{d}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}\cdot\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial n_{\mathrm{d}}}, \tag{B.96}$$

$$\tag{B.97}$$

with

$$\frac{\partial d}{\partial \eta} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}\eta}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\eta}, \tag{B.98}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \eta} = -\left[\mathit{\Delta}\lambda^{\mathrm{p}}\right]^{2}\xi_{\mathrm{m}}\,d\,f^{\boldsymbol{m}}\frac{1+\epsilon}{\left[f^{\boldsymbol{m}}+\epsilon\right]^{2}}\bar{\mathbf{v}}^{\mathrm{tr}} + \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\eta}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \eta}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.99}$$

$$\frac{\partial d}{\partial \xi_{\mathrm{vol}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}\xi_{\mathrm{vol}}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{vol}}}, \tag{B.100}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{vol}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{vol}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \xi_{\mathrm{vol}}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.101}$$

$$\frac{\partial d}{\partial \xi_{\mathrm{iso}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}\xi_{\mathrm{iso}}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{iso}}}, \tag{B.102}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{iso}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{iso}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \xi_{\mathrm{iso}}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.103}$$

$$\frac{\partial d}{\partial \xi_{\mathrm{q}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}\xi_{\mathrm{q}}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{q}}}, \tag{B.104}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{q}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{q}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \xi_{\mathrm{q}}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.105}$$

$$\frac{\partial d}{\partial \xi_{\mathrm{m}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}\xi_{\mathrm{m}}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{m}}}, \tag{B.106}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \xi_{\mathrm{m}}} = -\left[\mathit{\Delta}\lambda^{\mathrm{p}}\right]^{2}\eta\,d\,f^{\boldsymbol{m}}\frac{1+\epsilon}{\left[f^{\boldsymbol{m}}+\epsilon\right]^{2}}\bar{\mathbf{v}}^{\mathrm{tr}} + \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\xi_{\mathrm{m}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \xi_{\mathrm{m}}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.107}$$

$$\frac{\partial d}{\partial \eta_{\alpha}} = \left[\mathit{\Delta}\lambda^{\mathrm{d}}\right]^{2}f^{\alpha}\frac{1+\epsilon}{\left[f^{\alpha}+\epsilon\right]^{2}} + \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}\eta_{\alpha}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\eta_{\alpha}}, \tag{B.108}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \eta_{\alpha}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}\eta_{\alpha}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \eta_{\alpha}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.109}$$

$$\frac{\partial d}{\partial q_{\mathrm{min}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}q_{\mathrm{min}}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}q_{\mathrm{min}}}, \tag{B.110}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial q_{\mathrm{min}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}q_{\mathrm{min}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial q_{\mathrm{min}}}\bar{\mathbf{v}}^{\mathrm{tr}}, \tag{B.111}$$

$$\frac{\partial d}{\partial n_{\mathrm{d}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{d}}}{\mathrm{d}n_{\mathrm{d}}}\hat{f}^{\alpha} + \mathit{\Delta}\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}n_{\mathrm{d}}}, \tag{B.112}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial n_{\mathrm{d}}} = \frac{\mathrm{d}\mathit{\Delta}\lambda^{\mathrm{p}}}{\mathrm{d}n_{\mathrm{d}}}\hat{f}^{\boldsymbol{m}}\bar{\mathbf{v}}^{\mathrm{tr}} + \mathit{\Delta}\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial n_{\mathrm{d}}}\bar{\mathbf{v}}^{\mathrm{tr}}. \tag{B.113}$$

For the derivatives of the Lagriangian multipliers, following the implicit function theorem, the derivatives of the dissipation potentials read

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \eta}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2\,G\tilde{\varepsilon}\left[\frac{\partial f^{\mathrm{iso}}}{\partial \eta}\hat{f}^{\boldsymbol{m}} + f^{\mathrm{iso}}\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \eta}\right] - 2\,G\hat{f}^{\boldsymbol{m}} f^{\mathrm{iso}}\varDelta\lambda^{\mathrm{p}}\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \eta}, \tag{B.114}$$

$$\left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \eta}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -\frac{\partial q}{\partial \eta}\hat{f}^{\alpha} - q_{\mathrm{min}}\,n_{\mathrm{d}}\left[1-f^{q}\right]^{n_{\mathrm{d}}-1}\xi_{\mathrm{q}}\,df^{q}, \tag{B.115}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \xi_{\mathrm{vol}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0 \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \xi_{\mathrm{vol}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial \xi_{\mathrm{vol}}}\hat{f}^{\alpha}, \tag{B.116}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \xi_{\mathrm{iso}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -2\,G\,\eta\,d\,f^{\mathrm{iso}}\hat{f}^{\boldsymbol{m}}\tilde{\varepsilon} \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \xi_{\mathrm{iso}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \frac{\partial q}{\partial \xi_{\mathrm{iso}}}\hat{f}^{\alpha}, \tag{B.117}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \xi_{\mathrm{q}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0 \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \xi_{\mathrm{q}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -q_{\mathrm{min}}n_{\mathrm{d}}\left[1-f^{q}\right]^{n_{\mathrm{d}}-1}d\,\eta f^{q}, \tag{B.118}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \xi_{\mathrm{m}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2\,Gf^{\mathrm{iso}}\left[\tilde{\varepsilon} - \hat{f}^{\boldsymbol{m}}\varDelta\lambda^{\mathrm{p}}\right]\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \xi_{\mathrm{m}}}, \tag{B.119}$$

$$\left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \xi_{\mathrm{m}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2\,G\left[f^{\mathrm{iso}}\right]'\varDelta\lambda^{\mathrm{p}}\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \xi_{\mathrm{m}}}\tilde{\varepsilon}\hat{f}^{\alpha}, \tag{B.120}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial \eta_{\alpha}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 2\,G\varDelta\lambda^{\mathrm{p}}\frac{\partial \hat{f}^{\alpha}}{\partial \eta_{\alpha}}\left[\left[f^{\mathrm{iso}}\right]'\hat{f}^{\boldsymbol{m}}\tilde{\varepsilon} + f^{\mathrm{iso}}\left[\hat{f}^{\boldsymbol{m}}\right]'\left[\tilde{\varepsilon} - \hat{f}^{\boldsymbol{m}}\varDelta\lambda^{\mathrm{p}}\right]\right], \tag{B.121}$$

$$\left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial \eta_{\alpha}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \left[q - \hat{f}^{\alpha}\frac{\partial q}{\partial d}\varDelta\lambda^{\mathrm{p}} + q_{\mathrm{min}}\,n_{\mathrm{d}}\left[1-f^{q}\right]^{n_{\mathrm{d}}-1}\left[f^{q}\right]'\varDelta\lambda^{\mathrm{d}}\right]\frac{\partial \hat{f}^{\alpha}}{\partial \eta_{\alpha}}, \tag{B.122}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial q_{\mathrm{min}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0 \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial q_{\mathrm{min}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -q_{\mathrm{min}}\ln(1-f^{q})\left[1-f^{q}\right]^{n_{\mathrm{d}}}, \tag{B.123}$$

$$\left.\frac{\partial \varPhi^{\mathrm{p}}}{\partial n_{\mathrm{d}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0 \quad \text{and} \quad \left.\frac{\partial \varPhi^{\mathrm{d}}}{\partial n_{\mathrm{d}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = -\left[1-f^{q}\right]^{n}_{\mathrm{d}}. \tag{B.124}$$

with the additional partial derivatives

$$\begin{aligned}
\frac{\partial q}{\partial \eta} &= -\frac{1}{2} K \frac{\partial \left[f^{\mathrm{vol}}\right]'}{\partial \eta} \left[\varepsilon^{\mathrm{e,vol}}\right]^2 - G \frac{\partial \left[f^{\mathrm{iso}}\right]'}{\partial \eta} \tilde{\varepsilon}^2 + 2\, G \left[f^{\mathrm{iso}}\right]' \tilde{\varepsilon} \Delta\lambda^{\mathrm{p}} \frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \eta}, && \text{(B.125)}\\
\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \eta} &= \xi_{\mathrm{m}}\, d\, f^{\boldsymbol{m}} \frac{1+\epsilon}{\left[f^{\boldsymbol{m}}+\epsilon\right]^2}, && \text{(B.126)}\\
\frac{\partial \left[f^{\mathrm{vol}}\right]'}{\partial \eta} &= -\xi_{\mathrm{vol}}\, f^{\mathrm{vol}} - \xi_{\mathrm{vol}}\, d \left[f^{\mathrm{vol}}\right]', && \text{(B.127)}\\
\frac{\partial \left[f^{\mathrm{iso}}\right]'}{\partial \eta} &= -\xi_{\mathrm{iso}}\, f^{\mathrm{iso}} - \xi_{\mathrm{iso}}\, d \left[f^{\mathrm{iso}}\right]', && \text{(B.128)}\\
\frac{\partial q}{\partial \xi_{\mathrm{vol}}} &= \frac{1}{2} K \eta \left[f^{\mathrm{vol}} + d \left[f^{\mathrm{vol}}\right]'\right] \left[\varepsilon^{\mathrm{e,vol}}\right]^2, && \text{(B.129)}\\
\frac{\partial q}{\partial \xi_{\mathrm{iso}}} &= G\, \eta \left[f^{\mathrm{iso}} + d \left[f^{\mathrm{iso}}\right]'\right] \tilde{\varepsilon}^2, && \text{(B.130)}\\
\frac{\partial \hat{f}^{\boldsymbol{m}}}{\partial \xi_{\mathrm{m}}} &= \eta\, d\, f^{\boldsymbol{m}} \frac{1+\epsilon}{\left[f^{\boldsymbol{m}}+\epsilon\right]^2}, && \text{(B.131)}\\
\frac{\partial \hat{f}^{\alpha}}{\partial \eta_{\alpha}} &= \alpha\, f^{\alpha} \frac{1+\epsilon}{\left[f^{\alpha}\right]^2}. && \text{(B.132)}
\end{aligned}$$

The derivatives for the scalar nonlocal driving force can be computed using the quantities derived beforehand, such that

$$\begin{aligned}
\frac{\partial Y}{\partial \eta} &= -\beta_{\mathrm{d}} \frac{\partial d}{\partial \eta}, \quad \frac{\partial Y}{\partial \xi_{\mathrm{vol}}} = -\beta_{\mathrm{d}} \frac{\partial d}{\partial \xi_{\mathrm{vol}}}, \quad \frac{\partial Y}{\partial \xi_{\mathrm{iso}}} = -\beta_{\mathrm{d}} \frac{\partial d}{\partial \xi_{\mathrm{iso}}}, && \text{(B.133)}\\
\frac{\partial Y}{\partial \xi_{\mathrm{q}}} &= -\beta_{\mathrm{d}} \frac{\partial d}{\partial \xi_{\mathrm{q}}}, \quad \frac{\partial Y}{\partial \xi_{\mathrm{m}}} = -\beta_{\mathrm{d}} \frac{\partial d}{\partial \xi_{\mathrm{m}}}, \quad \frac{\partial Y}{\partial \eta_{\alpha}} = -\beta_{\mathrm{d}} \frac{\partial d}{\partial \eta_{\alpha}}, && \text{(B.134)}\\
\frac{\partial Y}{\partial q_{\min}} &= -\beta_{\mathrm{d}} \frac{\partial d}{\partial q_{\min}}, \quad \frac{\partial Y}{\partial n_{\mathrm{d}}} = -\beta_{\mathrm{d}} \frac{\partial d}{\partial n_{\mathrm{d}}}. && \text{(B.135)}
\end{aligned}$$

B.3.4 Regularisation parameters

Finally, the parameters contributing to the nonlocal regularisation have to be considered. However, the regularisation parameter c_{d} is only included in the nonlocal damage driving force $\boldsymbol{Y}$ and the derivatives already specified in Sect. 5.5. Therefore, only β_{d} has to be considered here.

$$\frac{\partial \bar{\mathbf{m}}}{\partial \beta_{\mathrm{d}}} = \frac{\partial \bar{\mathbf{m}}}{\partial d} \frac{\partial d}{\partial \beta_{\mathrm{d}}} + \frac{\partial \bar{\mathbf{m}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \beta_{\mathrm{d}}}. \qquad \text{(B.136)}$$

with the additional derivatives

$$\frac{\partial d}{\partial \beta_{\mathrm{d}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{d}}}{\mathrm{d}\beta_{\mathrm{d}}}\hat{f}^{\alpha} + \Delta\lambda^{\mathrm{d}}\left[\hat{f}^{\alpha}\right]'\sqrt{\frac{2}{3}}\frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\beta_{\mathrm{d}}}, \tag{B.137}$$

$$\frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \beta_{\mathrm{d}}} = \frac{\mathrm{d}\Delta\lambda^{\mathrm{p}}}{\mathrm{d}\beta_{\mathrm{d}}}\hat{f}^{\boldsymbol{m}}\bar{\boldsymbol{\nu}}^{\mathrm{tr}} + \Delta\lambda^{\mathrm{p}}\left[\hat{f}^{\boldsymbol{m}}\right]'\frac{\partial d}{\partial \beta_{\mathrm{d}}}\bar{\boldsymbol{\nu}}^{\mathrm{tr}}. \tag{B.138}$$

Again, the implicit function theorem is used, wherein the derivatives of the dissipation potentials read

$$\left.\frac{\partial \Phi^{\mathrm{p}}}{\partial \beta_{\mathrm{d}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = 0 \quad \text{and} \quad \left.\frac{\partial \Phi^{\mathrm{d}}}{\partial \beta_{\mathrm{d}}}\right|_{\lambda^{\mathrm{p}},\lambda^{\mathrm{d}}=\mathrm{const.}} = \left[\phi - d\right]\hat{f}^{\alpha}. \tag{B.139}$$

The derivative considering the scalar nonlocal driving finally yields

$$\frac{\partial Y}{\partial \beta_{\mathrm{d}}} = \phi - d - \beta_{\mathrm{d}}\frac{\partial d}{\partial \beta_{\mathrm{d}}}. \tag{B.140}$$

B.4 Compliance sensitivities

For the implementation of the compliance C into the optimisation process, the gradients are required and derived by its total variation, see (5.77). Therein, the partial derivatives of the compliance w.r.t. several quantities are not specified, which are derived here instead. The two elastic contributions to the local Helmholtz energy are combined into $\Psi^{\mathrm{el}} = \Psi^{\mathrm{vol}} + \Psi^{\mathrm{iso}}$ within the following equations. Since many of these upcoming equations utilise already established derivatives from the previous sections, their computation is straightforward.

The derivative w.r.t. the deformation gradient $\boldsymbol{F}$ is split, like established in (B.38), by utilising the chain rule, such that

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \boldsymbol{F}} = \frac{\partial \Psi^{\mathrm{el}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{F}}. \tag{B.141}$$

The latter derivative has been derived in (B.39), while the former results in

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} = \sum_{i=1}^{3}\frac{\partial \Psi^{\mathrm{el}}}{\partial \varepsilon_i^{\mathrm{e,tr}}}\frac{1}{2\left[\lambda_j^{\mathrm{e,tr}}\right]^2}\boldsymbol{P}_i. \tag{B.142}$$

The derivative therein can be decomposed into three additive terms, i.e.

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\varepsilon}^{\mathrm{e,tr}}} = \frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\varepsilon}^{\mathrm{e,tr}}} + \frac{\partial \Psi^{\mathrm{el}}}{\partial d}\frac{\partial d}{\partial \bar{\varepsilon}^{\mathrm{e,tr}}} + \frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\varepsilon}^{\mathrm{p}}} \cdot \frac{\partial \bar{\varepsilon}^{\mathrm{p}}}{\partial \bar{\varepsilon}^{\mathrm{e,tr}}} \tag{B.143}$$

with

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\varepsilon}^{\mathrm{e,tr}}} = \bar{\boldsymbol{m}}, \tag{B.144}$$

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial d} = \frac{K}{2} \left[f^{\mathrm{vol}}\right]' \left[\varepsilon^{\mathrm{e,vol}}\right]^2 + 2\, G \left[f^{\mathrm{iso}}\right]' \boldsymbol{\varepsilon}^{\mathrm{e,iso}} : \boldsymbol{\varepsilon}^{\mathrm{e,iso}}, \tag{B.145}$$

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} = -2\, G f^{\mathrm{iso}} \bar{\boldsymbol{m}}\, \bar{\boldsymbol{\nu}}^{\mathrm{tr}}, \tag{B.146}$$

and (A.11) and (A.12). Using these partial derivatives, as well as (A.13) and (A.14), the total derivative w.r.t. ϕ can be written as

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \phi} = \frac{\partial \Psi^{\mathrm{el}}}{\partial d} \frac{\partial d}{\partial \phi} + \frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \phi}. \tag{B.147}$$

Finally, the derivatives w.r.t. the history variables of the previous load step $\boldsymbol{h}_n$ can be decomposed into only known quantities, i.e.

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}} = \frac{\partial \Psi^{\mathrm{el}}}{\partial \boldsymbol{b}^{\mathrm{e,tr}}} : \frac{\partial \boldsymbol{b}^{\mathrm{e,tr}}}{\partial \boldsymbol{C}_n^{\mathrm{p}-1}}, \tag{B.148}$$

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial \alpha_n} = \frac{\partial \Psi^{\mathrm{el}}}{\partial d} \frac{\partial d}{\partial \alpha_n} + \frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial \alpha_n}, \tag{B.149}$$

$$\frac{\partial \Psi^{\mathrm{el}}}{\partial d_n} = \frac{\partial \Psi^{\mathrm{el}}}{\partial d} \frac{\partial d}{\partial d_n} + \frac{\partial \Psi^{\mathrm{el}}}{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}} \cdot \frac{\partial \bar{\boldsymbol{\varepsilon}}^{\mathrm{p}}}{\partial d_n}. \tag{B.150}$$

Appendix C

Additional results related to Part I

In this appendix, additional research results which relate to the academic challenges can be found.

C.1 Validation for the geometric optimisation

In [103] from the author of this thesis, shape optimisation was applied to a different regularised nonlocal damage model. The presented model therein is not derived here for the purpose of coherence. Instead, a parameter identification (PI) for the model in this thesis is conducted in Appendix C.1.1, which enables reproduction of the results. Nonetheless, in that work, the generated shapes were 3d printed with PLA and subsequently tested in micro tensile tests to qualitatively validate the presented work. Application of this approach to the results in the previous section are difficult. Since the model described in Sect. 4 is derived for metals, e.g. DP800, in mind and the optimisation conducted with those respective parameters, 3d-printed specimen with PLA cannot accurately represent the simulated results. While similar manufacturing methods for metals exist, its application to validate the previously generated structures is difficult. Selective Laser Melting (SLM) [89, 239] for example creates components by using metal powder and a laser to melt this powder at specific points on a 2d surface. After hardening the process is continued in a 3d space by printing further layers in height direction, similar to 3d printing with PLA. Since in this case the metal powder melts and solidifies, the material characteristics no longer coincides with the parameters used for sheet metals in this work. So, in theory, a validation is possible, but a different material model has be considered. Alternatively, the description by the underlying Computer Aided Geometric Design (CAGD) model could be simplified to enable machining of the optimised geometries, cf. [145, 146, 147], which allows further utilisation of the ductile damage model. However, this exceeds the scope of this work and therefore is not further followed.

C.1.1 Parameter identification

Since the material model in [103] only considers damage, the material model in this work can closely capture the same effects by disabling plasticity. Utilisation of the material sensitivities from Chap. 5 allows PI, aiming to reproduce the material behaviour from the cited work. In theory, the results from the shape optimisation [103] could be reproduced with this model, using the identified parameters. However, since this is not the aim of this work as metals are the considered material. The presented PI is only applied to allow a transfer of the methodology described in this work to the results in [103] and therefore mainly motivates the next section.

The optimisation problem utilises the objective function described in (5.81) and therefore reads

$$\underset{\mathbf{m}_{\mathrm{p,l}} \leq \mathbf{m}_{\mathrm{p}} \leq \mathbf{m}_{\mathrm{p,u}}}{\text{minimise}} \quad \mathfrak{J}^{\mathrm{F}} = \sum_{t=1}^{T} w_t \left[F_t^{\mathrm{sim}}(\mathbf{m}_{\mathrm{p}}) - F_t^{\mathrm{exp}} \right]^2 , \tag{C.1}$$

where $F_t^{\mathrm{sim}}(\mathbf{m}_{\mathrm{p}})$ are the forces from the model described in this work, and F_t^{exp} the forces from the model in [103]. Note, that the "experimental" sample points in this objective are not gathered from experiments but are the results of simulations with the material model of the cited work. The boundary value problem is identical, i.e. the plate with a hole, and the loading is prescribed accordingly. Two different sampling sizes are used and compared. A fine sampling of 100 load steps, which coincides with the original applied load steps, and a coarse sampling of only 10 steps are chosen. Since plasticity shall not be considered within the modelling here, it has to be disabled. How this accomplished for the model was already presented in Sect. 5.8 and Sect. 6.2.3. The weights $w_t = 1$ are chosen for simplicity.

The specific design variables to be fitted in this PI are listed in Table C.1, together with lower and upper bounds. The initial guess and optimal solutions for both sampling sizes are also depicted. The initial guess for this PI uses the parameter set for the model within [103]. The parameters which coincide with plasticity or the coupling between damage and plasticity are omitted within the optimisation. The same is true for the penalty parameter $\beta_{\mathrm{d}} = 500$, which is held constant. The damage rate factor $\eta = 1$ is chosen as such a parameter is not present in the cited work. Finally, the damage function $[1 - f^q]^{n_{\mathrm{d}}}$ is disabled in the damage dissipation potentials by setting $n_{\mathrm{d}} = 0$. Since the referenced work does not utilise such an approach, fitting with it enabled is not advisable. The optimisation is achieved using *lsqnonlin*, a solver within the Matlab toolbox explicitly intended to solve nonlinear least-square problems, such as (C.1).

Table C.1: Overview of the parameter identification.

$\mathbf{s}$	$\mathbf{s}_{\mathrm{l}}$	$\mathbf{s}_{\mathrm{u}}$	initial	optimum (10)	optimum (100)
E/MPa	0.5×10^3	1×10^3	0.7×10^3	0.7166×10^3	0.7166×10^3
ν	0.15	0.3	0.2	0.3	0.3
ξ_{vol}	0.5	2.0	0.9	0.5	0.5
ξ_{iso}	0.5	2.0	0.9	1.0924	1.0948
$q_{\mathrm{min}}/\mathrm{MPa}$	1	5	2.2	2.4333	2.4378
$c_{\mathrm{d}}/\mathrm{N}$	1	5	3	5	5

The resulting load displacement curves are depicted in Fig. C.1, together with the iterative process. While a perfect match could not be achieved, the results are nonetheless sufficiently close, resulting in a relative improvement compared to the initial guess of 99.04% for the coarse and 98.96% for the fine sample size. The relatively large remaining objective value in the fine sampled optimisation is a result of the large number of compared values, which results in the significantly lower value for the coarse problem. Since the coarse sampling leads to large linear segments in the load displacement curve between each computed load step, the solution is computed again with the same fine time load step size. However, since both identifications almost yield the same results, see Table C.1, the curves of the fine sampling and the recalculated solution in the load displacement curves almost coincide.

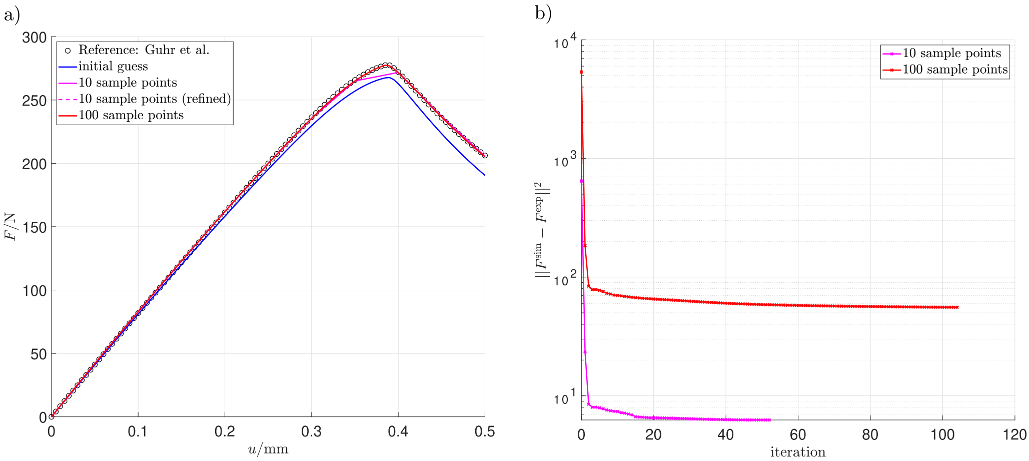


Figure C.1: Results of the parameter identification: a) The load displacement curves. b) The objective over iterations for both sampling sizes.

In addition to the numerical verification within Sect. 5.8, this successful PI with gradient-based methods further proofs the correct implementation of the derived material sensitivities. Naturally, the problem could be enhanced. For example, including the displacements $\mathbf{u}$ or simulated nonlocal damage fields ϕ within the objective function could further enhance the fitting. However, these enhancements are not performed here. This PI on the one hand only aims to show another application for the material sensitivities derived in Chap. 5 and on the other acts as a stepping stone to motivate the next section, and couple the results with the work presented in this thesis.

Remark C.1 The model in [103] uses a Fischer-Burmeister approach [76, 196] to solve the global KKT conditions due to the introduction of the nonlocal formulation. This is an alternative to the micromorphic approach presented in Chap. 4 and circumvents the need for a local damage formulation. This in turn no longer requires the history variation in Chap. 5 and therefore the updated optimisation framework. However, the global Fischer-Burmeister approach requires an additional global assembly of its residual. Even with the reduction in other computational requirements, i.e. the local damage model, the micromorphic approach is still more efficient than the global Fischer-Burmeister. The additional assembly of the global Fischer-Burmeister residual is computationally more expensive than the solving the local damage model.

C.1.2 Shape optimisation applied to nonlocal damage

For the validation, four optimisation problems are calculated, using the same objectives described in Sect. 5.6. For a damage optimisation the problems read

$$\begin{aligned} \underset{\mathbf{s}_\mathrm{l}\leq\mathbf{s}\leq\mathbf{s}_\mathrm{u}}{\text{minimise}} \quad & \mathfrak{J}^\mathrm{d} = \|\mathbf{d}(\mathbf{s})\|^2 \\ \text{subject to} \quad & V(\mathbf{s}) = V_0, \end{aligned} \tag{C.2}$$

and

$$\begin{aligned} \underset{\mathbf{s}_\mathrm{l}\leq\mathbf{s}\leq\mathbf{s}_\mathrm{u}}{\text{minimise}} \quad & \mathfrak{J}^\mathrm{d}\|\mathbf{d}(\mathbf{s})\|^2 \\ \text{subject to} \quad & V(\mathbf{s}) = V_0 \\ & d_i(\mathbf{s}) \leq d_\mathrm{crit}. \end{aligned} \tag{C.3}$$

For a compliance optimisation, the problems

$$\begin{aligned} \underset{\mathbf{s}_\mathrm{l}\leq\mathbf{s}\leq\mathbf{s}_\mathrm{u}}{\text{minimise}} \quad & \mathfrak{J}^\mathrm{C} = C(\mathbf{s}) \\ \text{subject to} \quad & V(\mathbf{s}) = V_0, \end{aligned} \tag{C.4}$$

and

$$\begin{aligned} \underset{\mathbf{s}_\mathrm{l}\leq\mathbf{s}\leq\mathbf{s}_\mathrm{u}}{\text{minimise}} \quad & \mathfrak{J}^\mathrm{C} = C(\mathbf{s}) \\ \text{subject to} \quad & V(\mathbf{s}) = V_0 \\ & d_i(\mathbf{s}) \leq d_\mathrm{crit}, \end{aligned} \tag{C.5}$$

are optimised. Since in this model the damage variable d is a nonlocal quantity, it exchanges the ϕ in the model presented throughout this work. The CAGD and boundary value problem are nearly the same as in Fig. 6.1 and Fig. 6.2. However, the design space in thickness direction is restricted such that a change in thickness is prohibited. This eases with the 3d-printing of the specimen and therefore the validation. The resulting shapes are presented in Fig. C.2 for the damage optimisation and Fig. C.3 for the compliance optimisation.

Comparing the results of the damage optimisation to the ones in Sect. 6.2.2 shows a large discrepancy for the unconstrained optimisation. The purely damage optimisation does not show the excessive localisation behaviour of (6.1) in Fig. 6.3a). While a small amount of localisation of damage occurs in the lower left part of the plate, it is not as pronounced. The damage constrained optimisation however, is very similar to Fig. 6.3d), i.e. the lower part of the plate is very large, while the plate gets thinner in the middle. Furthermore, the right-side edge is curved similarly. Only the left side shows a different behaviour, wherein the new design has a minimised width on the left edge of the plate.

While no compliance optimisation with damage constrained was computed in Sect. 6.2.3, the volume constrained optimisation (C.4) is quite similar. Both solutions straighten the lower part of the plat and remove a lot of curvature in the central hole. This curvature is pushed upwards, limited by the bounds of the respective design variables. This creates an elliptic hole in the centre. The optimisation problem (C.5) includes an additional nodal damage constraint. The resulting geometry therefore is more akin to the damage optimisation of (C.3), i.e. the outer and inner edges of the plate are curved. However, to increase the stiffness, the lower plate is not shifted to the right and additionally the overall cross-section is maximised.

C.1.3 Experimental data from 3d-printed samples

With the optimised geometries at hand, physical specimens are created by means of 3d printing with PLA. Thereby, the simulated response is validated with experimental data. A photo of the specimens is displayed in Fig. C.4a). The physically printed specimens match the numerically determined geometry, but have a longer section attached to the discretised domain in order to clamp the specimens in the micro-tension device, see Fig. C.4b). The specimens are loaded by displacement controlled monotonous tension. Due to the limitations of the material model considered at this stage, the accurate behaviour of PLA cannot be reproduced exactly in the simulations. The material parameters used during the optimisation are manually fitted by using the experimental data of the reference geometry in such a way that the point of damage initiation and evolution of damage can be predicted, see Fig. C.5a). The experimental specimens are generated from the optimised CAGD by exporting `.stl`-files — a file-format often used in 3d-printing — which are then used to generate the data for the 3d printer.

During optimisation, the maximum prescribed displacement in the simulation is $u^{\text{pre}} = 0.5\,\text{mm}$. However, the samples in the experiments are loaded until failure. In order to enable a proper comparison, the simulations are continued until a prescribed displacement of $u^{\text{pre}} = 1.5\,\text{mm}$ is reached.

The results for the comparison of the damage optimised sample is presented in Fig. C.5b). The validation of the geometries obtained from the damage minimisation focuses solely on the geometry resulting from the optimisation with constrained damage, since both shapes

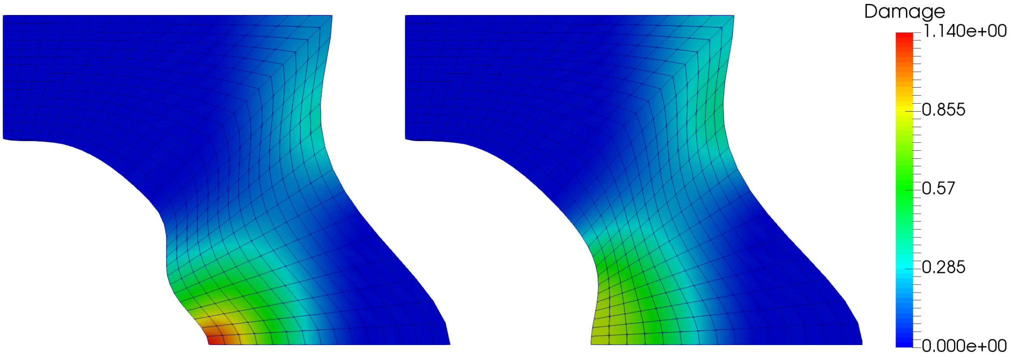


Figure C.2: Optimised geometries for the two approaches of two objective functions related to damage minimisation: The volume constrained on the left and the volume and damage constrained on the right. The contour plots show the undeformed design with the damage state after load application.

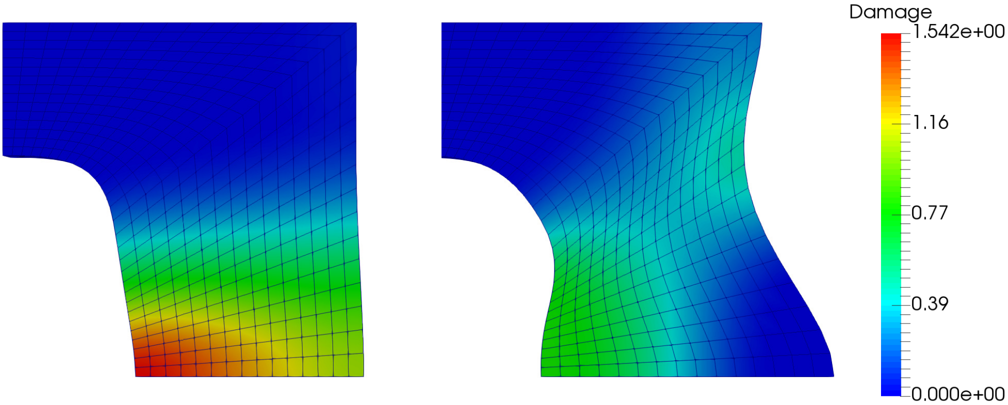


Figure C.3: Results for the compliance optimisation. The left contour plot shows the volume constrained problem, whereas the right is based on the objective function that constrains damage to a maximum value of $d_{\mathrm{crit}} = 0.8$.

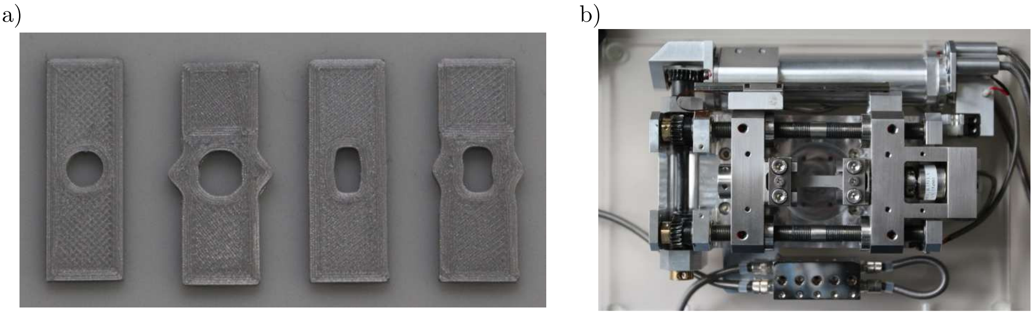


Figure C.4: a) 3d-printed specimens. The geometries from left to right: Reference, damage optimised with volume and damage constraint, compliance optimised with volume constraint, and compliance optimised with volume and damage constraint. b) Micro tensile machine used for the experiments of the 3d-printed specimens.

of the optimised geometries from damage minimisation only differ slightly. Qualitatively, the response of the simulation shows a good agreement with the experimentally obtained result. Especially in comparison to the reference geometry, the same trend in the force-displacement response can be observed. The stiffness is reduced slightly, while the point of peak force is reached at higher (displacement) loading level. In addition, the force peak becomes less sharp, albeit not as much as in the experiments. The small force plateau observed in the experimental data can most likely be attributed to plastic effects which, however, are not considered in the simulation.

The compliance optimised results in Fig. C.6 show the desired increased stiffness. The experimental data of the test with the geometry corresponding to the compliance optimisation without the damage constraint in Fig. C.6a) displays an even higher stiffness than predicted by the simulation. However, while the maximum reaction forces are drastically increased, damage seemingly starts to evolve very quickly leading to fracture

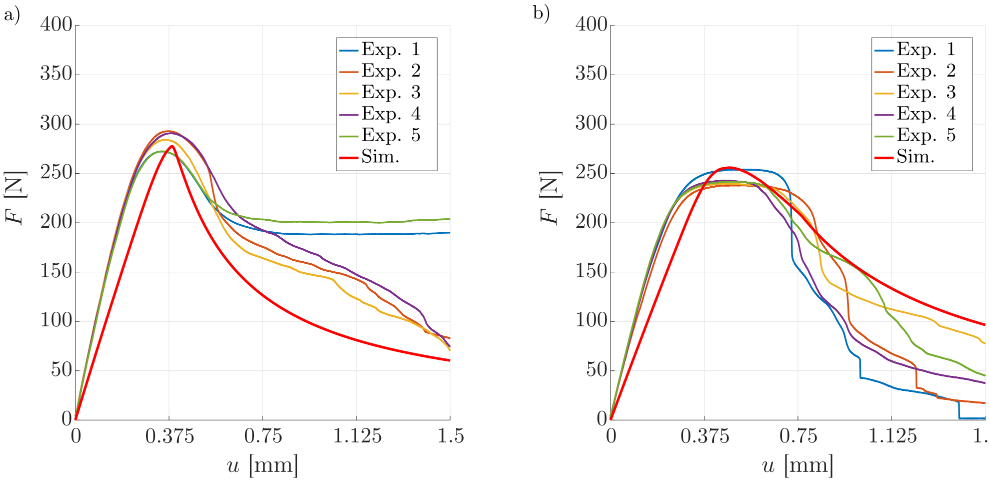


Figure C.5: Experimental results for the validation with the corresponding simulation displayed in red. a) The results for the reference geometry and b) the results for the damage optimisation with the additional damage constraints.

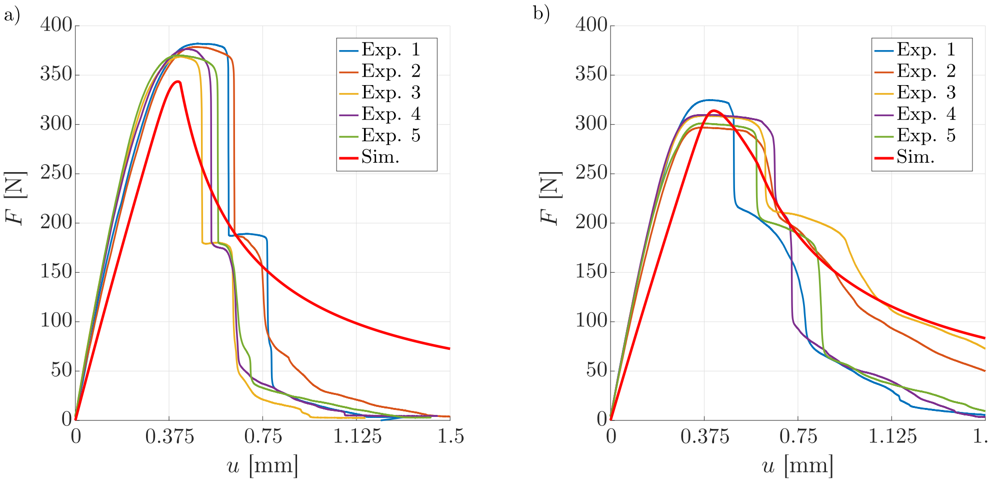


Figure C.6: Experimental results for the validation of the compliance optimisation. a) The results for the minimisation with only a volume constraint and b) with the additional damage constraints.

shortly after damage initiation. This cannot be directly predicted by the model, where only a steady loss of stiffness can be modelled.

Concerning the tendency of the force-displacement response in comparison with the reference geometry, a good agreement between experiments and simulation for the geometry with optimised compliance subject to a damage constraint is obtained, see Fig. C.6b). The initial stiffness of reference and optimised geometry nearly coincide and the maximum reaction forces also reach a similar value. However, the plateau observed in the experimental response cannot be predicted with this model. Overall, the comparison between reference and optimised geometries reveals — with limited applicability of the material model towards PLA — a good validation of the optimisation response. The analysed optimised geometries all confirm the tendencies of the simulations.

C.2 Raw material sensitivity data

This section includes additional data for the material parameter study in Sect. 6.3. Fig. C.7 and Fig. C.8 depict the raw sensitivity data for the notched plate and the plate with a hole, respectively. Fig. C.9 and Fig. C.10 depict the respective load paths, i.e. load step over applied external displacement.

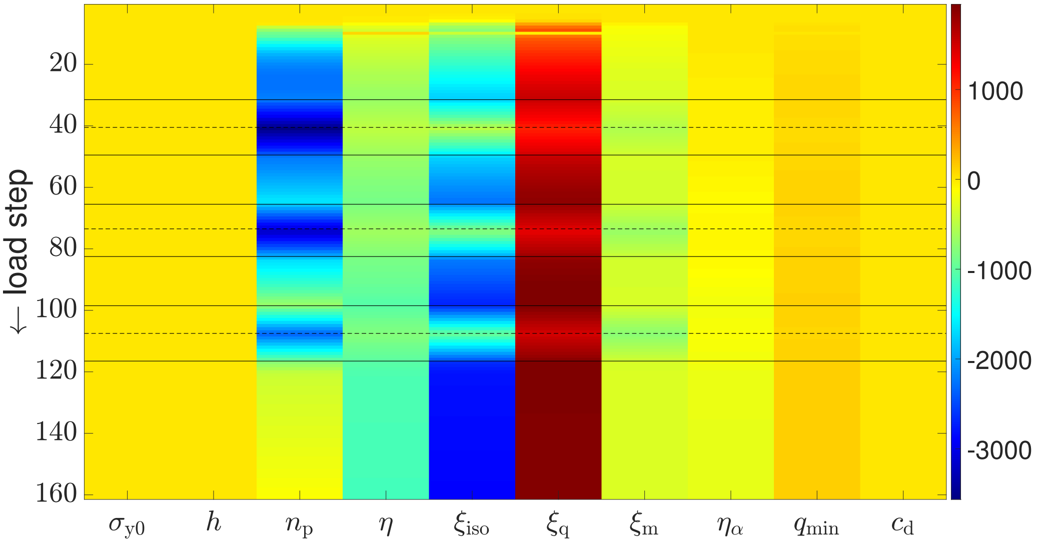


Figure C.7: Raw material sensitivity data for the notched plate. Load steps increasing from top to bottom.

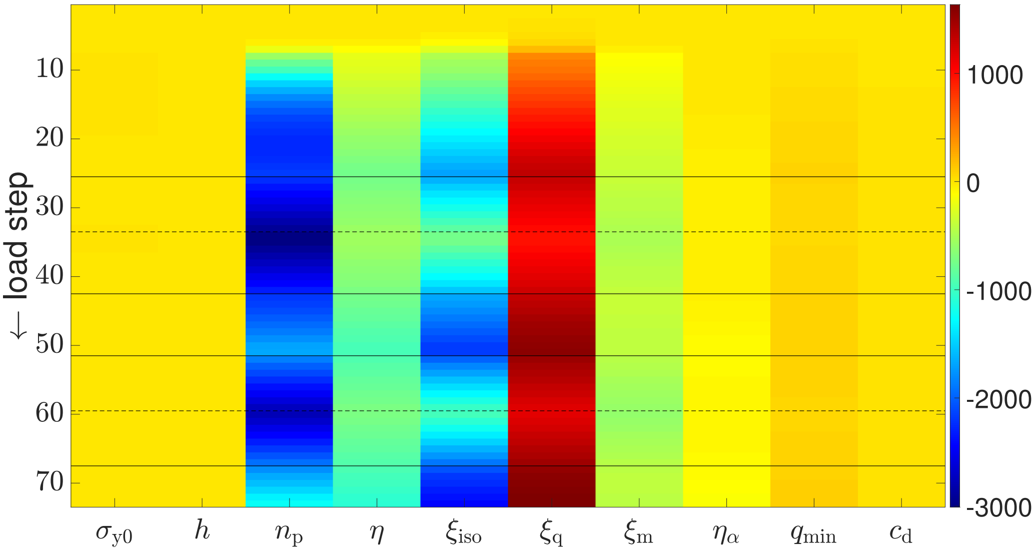


Figure C.8: Raw material sensitivity data for the plate with a hole. Load steps increasing from top to bottom.

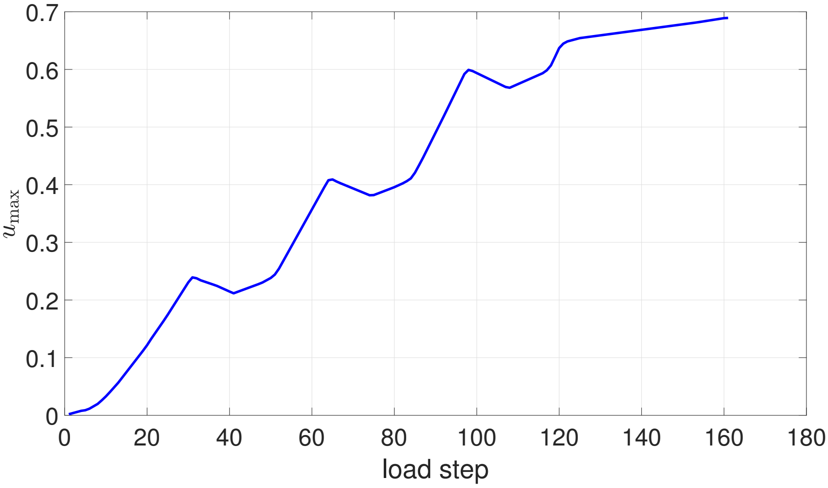


Figure C.9: Applied iterative maximal displacement over time load steps for the notched plate.

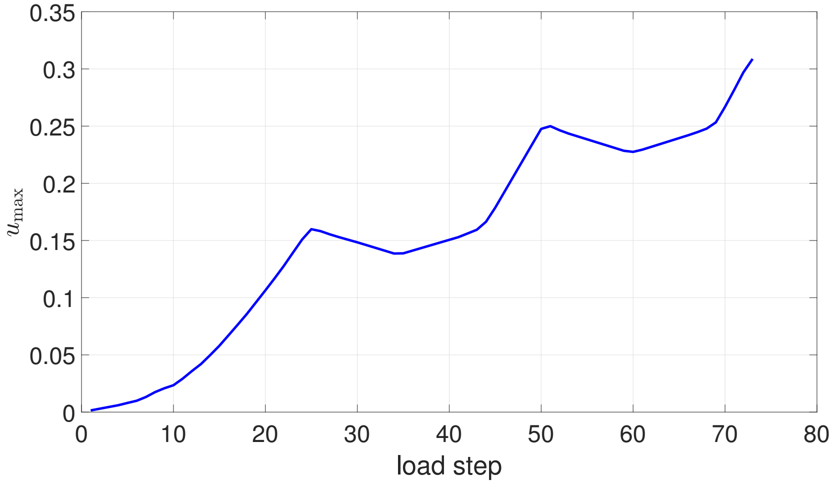


Figure C.10: Applied iterative maximal displacement over time load steps for the plate with a hole.

Appendix D

Additional results related to Part II

In this appendix, further optimisation results for all three optimised forming processes in Chap. 10 can be found.

D.1 Rod extrusion

In order to validate the results for the optimisation of the forward hollow extrusion process, experiments are to be conducted. However, due the limited number of available tools for the underlying process, an additional optimisation problem is stated. Therein, the extrusion strain is upper bounded by the value of $\varepsilon_{\mathrm{ex,u}} = 1.0$, since most of the available tool sets can achieve comparable extrusion strains close to this value. The optimisation problem for the experimental validation thus reads

$$\begin{aligned} &\underset{\mathbf{s}_\mathrm{l} \leq \mathbf{s}_2 \leq \mathbf{s}_\mathrm{u}}{\text{minimise}} && \mathfrak{J}^{\mathrm{re}} = \eta^{\mathrm{mean}}(\mathbf{s}_2) \\ &\text{subject to} && 0.2 \leq \varepsilon_{\mathrm{ex}}(r_{\mathrm{m}},r_1) \leq 1.0, \end{aligned} \tag{D.1}$$

with the same set of design variables as the previous optimisation. The initial parameters are chosen according to available tool sets, i.e. $r_1 = 13.575\,\mathrm{mm}$, $r_{\mathrm{m}} = 9.5\,\mathrm{mm}$ and $2\alpha = 60°$, respectively. The optimal process parameters generated for the above problem and the given initial value are listed in Table D.1. However, the resulting parameters are not explicitly convertible to available tool sets. As a result, close sets have to be chosen for the comparison of the initial and optimal solution within the experiments. The approximated parameters are listed under "approx." in Table D.1. The shoulder angle 2α and final extrusion radius r_1 are similar to the generated results. However, the mandrel radius is significantly larger by $r_{\mathrm{m}}^{\mathrm{appr}} = 0.8\,\mathrm{mm}$, which leads to a larger extrusion strain of $\varepsilon_{\mathrm{ex}}^{\mathrm{appr}} = 1.076$. Nonetheless, since only the optimisation is to be validated qualitatively, the approximately chosen tool sets should suffice.

The extrusion experiments are performed using a hydraulic press of model HZPUJ 260/160-1000/100 by manufacturer SMG at room temperature, employing a punch velocity of $10\,\mathrm{mm/s}$. Billets made of 16MnCrS5 are machined to an outer diameter $d_0 = 30\,\mathrm{mm}$, and the inner hole is drilled to accommodate the mandrel's diameter. Subsequently, the billets underwent sandblasting and are coated with a $\mathrm{MoS_2}$-containing lubricant Bechem Beruforge 191. Following extrusion, the produced parts are ejected using an ejector punch. The manufactured parts are shown in Fig. D.1a).

Table D.1: The parameters for the optimisation concerning the validation. The last column depicts the choosen tools which are available and close to the optimised results in column four.

s	dependent	fixed	initial	optimal	approx.
-	-	r_0/mm	15.1	15.1	15.1
r_1/mm	-	-	13.575	12.585	12.9
r_m/mm	-	-	9.5	10.841	11.6
-	ε_ex	-	0.382	0.991	1.076
2α/°	-	-	60	37.855	30

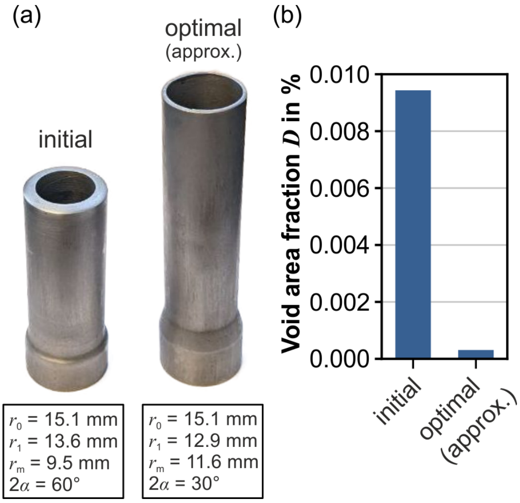


Fig. D.1: (a) Manufactured parts with initial parameter set as well as the tool set at hand with values closest to the optimised parameter set and (b) corresponding void area fractions measured by SEM investigations.

To quantify void areas, the parts are longitudinally cut and examined through scanning electron microscopy (SEM). A comprehensive approach involving both SEM and automated energy dispersive X-ray spectroscopy (EDX), detailed in [110], is employed to acquire data on void area fractions. The EDX method facilitates the differentiation between voids and materials, such as Al_2O_3-particles, which exhibited similar grayscale values in SEM images. To ensure statistically significant measurement areas, a total of 450 individual micrographs are scanned, resulting in a measurement area of $1.28\,\mathrm{mm}^2$ for each measurement. These areas for measurements are located at the inner radius of the extrudate, aligning with the positions assessed in numerical investigations, see Fig. 10.1). For the extrudate with the initial parameter set, a grid of 15 by 30 micrographs is utilized, while a grid of nine by 50 micrographs was used for the part representing the optimised parameter set to maintain a proportional coverage of the wall thickness. These experiments and measurements were conducted by Robing Gitschel at IUL, TU Dortmund University; for additional details, see [90].

The void area fraction demonstrates a significant reduction, decreasing from 0.0094% with the initial tool set to 0.00029% for tools representing the optimal parameter set, as illustrated in Fig D.1b). This reflects a substantial reduction of 96.14% in void area fraction from the initial to the optimal set. Consequently, it can be concluded that the optimisation results effectively capture experimental data, underscoring the advantages of employing the presented framework.

D.2 Bending

In this section, additional optimisations for the process of free air bending are presented. These encompass load and load path optimisation. This requires small adjustments to the simulation model for the same forming process presented in Sect. 9.3. An area on the surface has to be defined where the external loads are applied. This area for the load application has a length of $x_{\text{load}} = 1.78\,\text{mm}$, starting from $x = y = 0$ in the local coordinate system as depicted in red in Fig. D.2. Only the nodes on the surface of the outer bending area of the sheet are chosen for load application.

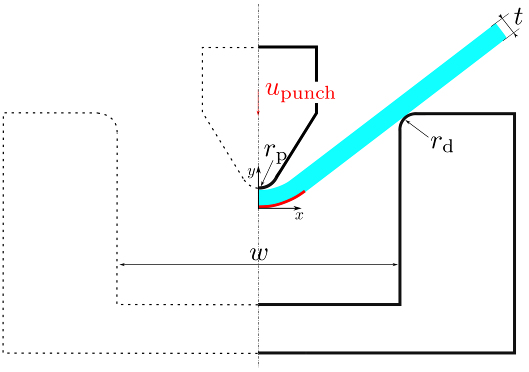


Fig. D.2: Sketch of the simulation model for the bending optimisations. The red line on the sheet represents the area of load application. Only the right side of the process is simulated.

D.2.1 Load optimisation

The first additional application of numerical optimisation for the bending process is external load optimisation. The basic idea is to exchange the elastomer cushion within elastomer bending, see Fig. 10.6b), and apply external loads to the outer area of the sheet, see Fig. 10.6c). These external loads can for example simulate the forces resulting from pressing the sheet into the elastomer cushion.

To apply external loads, two sets of polynomials $f_i = a_i x^2 + b_i x + c_i$ in x- and y-direction, respectively, are defined, cf. [95, 97, 101, 148]. Herein, x describes the local coordinate system for the bending process, originating in the lower left corner of the sheet within the simulation, see Fig. D.2. The coefficients a_i, b_i and c_i act as design variables $\mathbf{s}$ for the optimisation and hence govern the shape of the external loads. To map the computed loads to the Abaqus simulation, two force sets representing the x- and y-directional loads are defined at each of the selected nodes. The values from the polynomials are thus applied to these nodal force values during each optimisation step, allowing changes to the external forces in each iteration. While this is the general approach, for the following optimisation results certain boundary conditions are applied to the utilised polynomials. First of all, the loads are forced to be perpendicular at the coordinate origin (0,0), i.e. the left-most area of the simulated sheet. This can easily be implemented by fixing $c_x = 0$, which results in a force of $f_x(x = 0) = 0$ and therefore only a force in y-direction can be generated throughout the optimisation. Additionally, the parabolic parameter a_x is set to zero to force a linear force development. The force in y-direction is forced to be parabolic, thereby restricting the parameter $b_y = 0$. These constraints to the variables aim to generate a smooth force distribution to improve the emulated a reaction force due pressing the sheet into an elastomer cushion. An overview and the results are displayed in Table D.2.

Table D.2: An overview of the design and the fixed variables for load optimisation. Additionally, their initial and optimal values.

$\mathbf{s}$	fixed	initial	optimal
-	a_x	0	0
b_x	-	0	-3.183
	c_x	0	0
a_y		0	-0.7377
-	b_y	0	0
c_y	-	0	3.495

Since only an elasto-plastic model is used to simulate the material of the sheet, the triaxiality measurement (8.4) is chosen to quantify and optimise the damage. Furthermore, the results from [221] are chosen as a referential experiment. The improvements regarding triaxiality distribution and damage reduction with elastomer bending are used to define the optimisation problem. Furthermore, since the basic idea is to form the same part with reduced damage within the adjusted forming process, the geometry of initial and optimal shape must coincide, or rather, the difference shall be minimised. With these two criteria, the optimisation is defined as

$$\begin{aligned} \underset{\mathbf{s}_\mathrm{l} \le \mathbf{s} \le \mathbf{s}_\mathrm{u}}{\text{minimise}} \quad & \mathfrak{J}^{\mathrm{Load}} = \|\mathbf{u}(\mathbf{s}) - \mathbf{u}_{\mathrm{pre}}\|^2 \\ \text{subject to} \quad & \eta_i \le \eta_{\mathrm{crit}}. \end{aligned} \tag{D.2}$$

In this problem $\mathbf{u}(\mathbf{s})$ represents the nodal displacements of the sheet depending on the design variables $\mathbf{s}$, $\mathbf{u}_{\mathrm{pre}}$ the nodal displacements of the sheet without application of any external load and η_i is the vector of triaxiality values of the nodes which lie within the selected area of the sheet. The critical triaxiality value $\eta_{\mathrm{crit}} = 0.48$ is chosen, based on the results from [221]. Currently, only the triaxiality stress state after forming is considered for the optimisation, i.e. after maximal punch displacement $u_{\mathrm{punch}} = 5\,\mathrm{mm}$. In order to allow an optimisation with these external loads, the area which is considered within the following optimisation has to be smaller in width than for the applied loads x_{load}. As such, the area where the triaxiality values for the optimisation are evaluated is only $x_\eta = 1.5\,\mathrm{mm}$ wide.

The optimisation problem is solved using SQP. Due to the low number of contact interactions, compared to for example rod extrusion in the previous section, the numerical gradients are accurate enough to enable a quickly converging solution. This was also presented to be possible in the benchmark problems in Sect. 9.3. The initial state and optimal load distribution are depicted in Fig. D.3. Initially, no external loads are applied, i.e. free air bending like in Fig. 10.6a) is simulated. The triaxiality is largely homogenous and exceeds the desired threshold considerably, leading values close to $\eta_{\max} = 0.58$. In the optimal setting, the external forces are able to reduce the triaxiality value in the optimised area below the applied constraint. While at $x = 0$ the forces are perpendicular, as prescribed in the optimisation problem, the resulting force vector at each node onwards in x-direction is more slanted towards the left. This is a result of the enforced linear nature of the forces in x-direction. The total forces increase when approaching the right boundary of the surface, see Fig. 10.6. This is expected, since fewer total nodal forces apply to the nodes on the right and therefore more individual load is necessary to reduce the triaxiality.

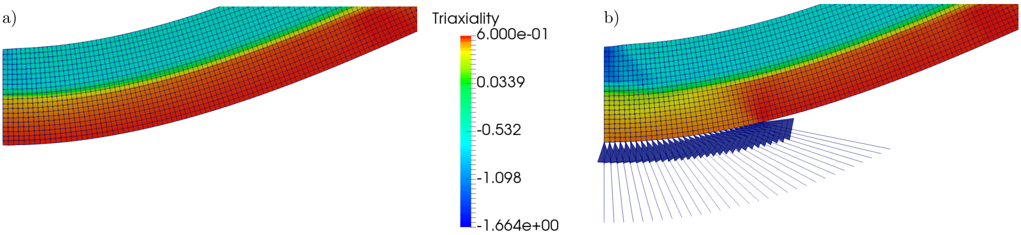


Figure D.3: a) Initial and b) optimal external forces for the load optimisation of free air bending.

The applied load optimisation is very basic in nature but is thought off as a proof of concept. Extending the optimisation to enable more complex load forms, or splitting the loads into different sections which allow more design freedom, could be implemented. However, this basic optimisation is enhanced in the next section, by controlling not only the final triaxiality value, but controlling its behaviour throughout the forming process. Another extension to this concept could be the inclusion of constraints regarding the direction of external loads, e.g. enforcing perpendicular loads w.r.t. the surface of the outer sheet.

An important note regarding the choice of initial design within load optimisation has to emphasised. For certain starting guesses, the optimisation might run into trouble with application of too high external loads. In this case the external loads will push with enough force, that the sheet metal will lose contact to the sheet holder on the right side. This will start a cascading effect which will end up with the force getting larger with each subsequent iteration. As a consequence, the sheet metal is wrapped around the die and the simulation is no longer able to converge. Therefore, it is vital to be careful regarding bounds of the applied loads as well as the initial guess.

D.2.2 Load path optimisation

In the previous optimisation the final triaxiality value in the sheet after forming was optimised. Consequently, only the last load step of the forming process is taken into account for the optimal load. However, the simple optimisation in Sect. D.2.1 might not result in an optimally damage reduced forming process. It is postulated, that the arbitrary critical triaxiality value η_{crit} might lead to a unsatisfactory amount of damage evolution. Consequently, if that value is exceeded during any time step within the process damage might evolve beyond an acceptable level. To circumvent this shortcoming, in this section, the intermediate triaxiality values during the forming process, and therefore the load path itself, are optimised.

The chosen nodes, and the problem exemplary for respective node A, are depicted in D.4. As can be identified, early during the displacement of the punch the triaxiality exceeds the critical triaxiality value $\eta_{\text{crit}} = 0.48$ in node A. Only after approximately $u = 2.75\,\text{mm}$

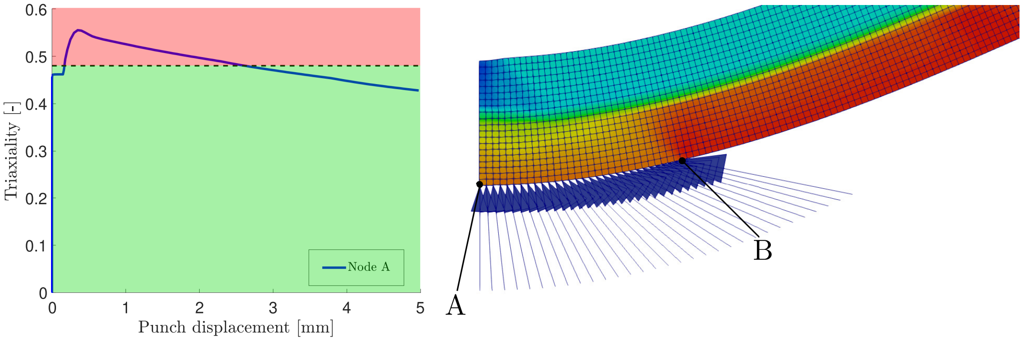


Figure D.4: The motivation for the optimisation of the nodal load paths on the left. The red area indicates a value of critical triaxiality exceeded. The right shows two different locations A and B which are optimised.

does the triaxiality value decrease below the threshold. Similar behaviour can be detected for other nodes. However, only node A and B are of interest in the following. They are chosen since they lie at either edge of the optimised area regarding force application, therefore should show a vastly different optimised load path behaviour.

To optimise the respective nodal triaxiality behaviour, the forces are loaded nonlinearly. This can be achieved in Abaqus by adjusting the respective amplitude values, i.e. the load application over simulation time. These amplitudes are described by B-splines within the optimisation setup. The control points of the B-splines then act as the design variables of the optimisation. For each problem, 6 control points are chosen equidistantly in x-direction, i.e. on the time-scale. The first control point remains fixed at [0,0], while the remaining two may reach any value between [0,1] in y-direction. The generated curves are inserted into the amplitude values of the simulation. Since a pseudo-time interval $t = [0,1]$ is chosen, the respective amplitudes have to be restricted to this time interval. The optimisation problem is similar to the previous one, i.e. the nodal displacements $\mathbf{u}(\mathbf{s})$ after prescribed punch displacement $\mathbf{u}_{\text{pre}}$ are compared. In addition, now the constraints are triaxiality values throughout the process for the respective chosen node. The optimisation problem thus reads

$$\begin{aligned} \underset{\mathbf{s}_{\text{l}} \leq \mathbf{s} \leq \mathbf{s}_{\text{u}}}{\text{minimise}} \quad & \mathfrak{J}^{\text{Amp}} = \|\mathbf{u}(\mathbf{s}) - \mathbf{u}_{\text{pre}}\|^2 \\ \text{subject to} \quad & \eta_j^{A/B}(t) \leq \eta_{\text{crit}}. \end{aligned} \tag{D.3}$$

While this optimisation problem reads very similar to (D.2), in the above the nodal triaxiality values $\eta_j^{A/B}$ only take one node, either A or B, into consideration. However, they now span a vector of nodal values over the simulation time t, with $j = 1, \ldots, n_{\text{lstp}}$ and n_{lstp} the number of load steps.

The results for the two different nodes are depicted in Fig. D.5. Each column illustrates the initial, linear, as well as the optimised, nonlinear amplitude and the resulting initial and optimal triaxiality history for their respective node. The optimisation allows the nodal triaxiality values to stay below the critical threshold throughout the whole process. While

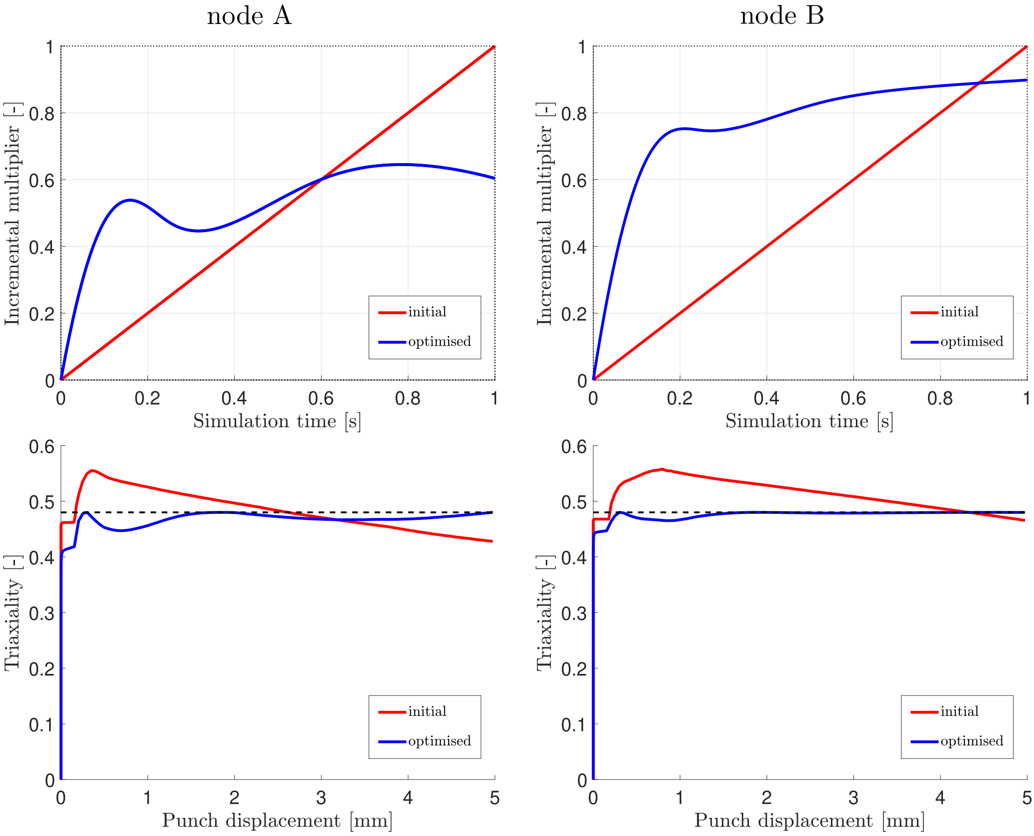


Figure D.5: Initial and optimal load paths for both nodes in the top, and below the respective triaxiality over the punch displacement.

node A reaches a triaxiality value below the threshold early during the linear loading, i.e. at an approximate punch displacement $u_{\text{punch}} = 2.75\,\text{mm}$, node B only just achieves this value shortly before forming is finished, i.e. $u_{\text{punch}} = 4.5\,\text{mm}$. With the adjusted load paths both nodes stay below the critical threshold throughout the complete process. Since the forces from the external loads in the previous section are not perpendicular with respect to the bended sheet, but instead steer more towards the left, node A requires a lower total force to reduce the triaxiality load path. More total force is applied to this node, which in turn reduces the triaxiality further. Consequently, the incremental multiplier for node A can be significantly reduced overall, with it reaching a value of 0.6 at the end of the forming process for example. For node B, since the total force applied to this node is lower, the multiplier is increased significantly more throughout the whole process, reaching a value of 0.9 at the end of the forming process.

Currently the presented method is only applied to each node individually, i.e. one node is chosen and the load path is optimised for the chosen node and therefore the complete applied forces are adjusted by the amplitude changes. An extension of this concept should allow the optimisation of multiple nodal behaviours. Additionally, the triaxiality curves show a very winding behaviour. While this is only significant for node B in the beginning, after which the curve flattens out, this is very noticeable for node A. This could be fixed by increasing the amount of control points which govern the B-spline, or by segmenting the complete amplitude into multiple B-splines, which allows for more design freedom within the optimisation. In [161], a process was developed which emulates this kind of behaviour. Instead of applying a constant load on the outer bending area, the external forces are time and process dependent, which is similar to the loads optimised above.

D.3 Stretch indenting

This section lists the explicit values for the optimisation of stretch indenting. Due to the size of the data, their explicit listing was omitted in Sect. 10.4

Table D.3: The initial parameters for stretch indenting and resulting geometries of the drawbead. Additionally, they are categorised according to the description Sect. 10.4.

label	r_1	r_2	h_s	y_s	x_s	type
x01	7	9	4.00	3.00	18.65	random value
x02	4	5	1.00	3	13.42	random value
x03	9	2	6.90	2.1	12.93	random value
x04	9	3	3.00	6	20.78	random value
x05	2.01	2	0	2.009	6.95	smallest bead
x06	9	9	5.20	3.8	22.12	largest bead
x07	5	7	0.50	4.5	18.73	average bead
x08	9	5	6.99	2.01	14.46	improved guess
x09	7	5	4.99	2.01	13.30	improved guess
x10	4	5	1.99	2.01	11.34	improved guess
x11	2.5	5	0.49	2.01	10.22	improved guess

Table D.4: The optimal design variables for stretch indenting and resulting parameters for optimisation (10.10). Additionally, the initial and optimal objectives.

label	r_1	r_2	h_s	y_s	x_s	$\mathfrak{J}^{\text{max}}_{\text{init}}$	$\mathfrak{J}^{\text{max}}_{\text{opt}}$
x01	6.3541	8.9946	4.35	15.17	2.0041	0.29196	0.25926
x02	3.0777	5.0903	1.077	10.71	2.0007	0.29201	0.26267
x03	8.9946	2	6.9844	12.67	2.0102	0.26849	0.26009
x04	8.8608	2.1132	6.8517	12.66	2.0091	0.32883	0.26003
x05	2.0163	2.0074	0.001	6.97	2.0153	0.26927	0.26696
x06	8.9925	8.9997	5.3177	21.79	3.6748	0.30052	0.29507
x07	2.5144	8.9622	0.5093	12.96	2.0051	0.30883	0.26165
x08	8.9999	5.2602	6.9842	14.62	2.0157	0.26109	0.25971
x09	4.0008	5.4862	1.9899	11.68	2.0109	0.26478	0.26028
x11	2.4937	5.3442	0.4909	10.47	2.0028	0.26438	0.26182

Table D.5: The optimal design variables for stretch indenting and resulting parameters for optimisation (10.11). Additionally, the initial and optimal objectives.

label	r_1	r_2	h_s	y_s	x_s	$\mathfrak{J}_\mathrm{init}^\mathrm{norm}$	$\mathfrak{J}_\mathrm{opt}^\mathrm{norm}$
x01	6.6569	8.9972	4.6163	15.46	2.0406	0.94187	0.81964
x02	3.0145	5.2091	1.0062	10.77	2.0083	0.94452	0.81988
x03	8.9918	2.0083	6.9569	12.75	2.0349	0.8486	0.81651
x04	8.9411	3.0115	6.9342	13.26	2.0069	1.20318	0.81195
x05	2.0029	2.0026	0.001	6.94	2.0019	0.84948	0.84565
x06	8.9572	8.965	6.9464	16.5	2.0108	0.98137	0.81329
x07	2.5238	8.9453	0.5158	12.97	2.008	1.0217	0.82258
x08	9	5.3444	6.9886	14.65	2.0114	0.81681	0.81205
x09	7.0075	5.7347	4.9884	13.77	2.0191	0.82016	0.81369
x10	4.0049	7.5068	1.9895	13.01	2.0154	0.82837	0.81853
x11	2.4962	5.3983	0.4907	10.52	2.0055	0.82983	0.82326