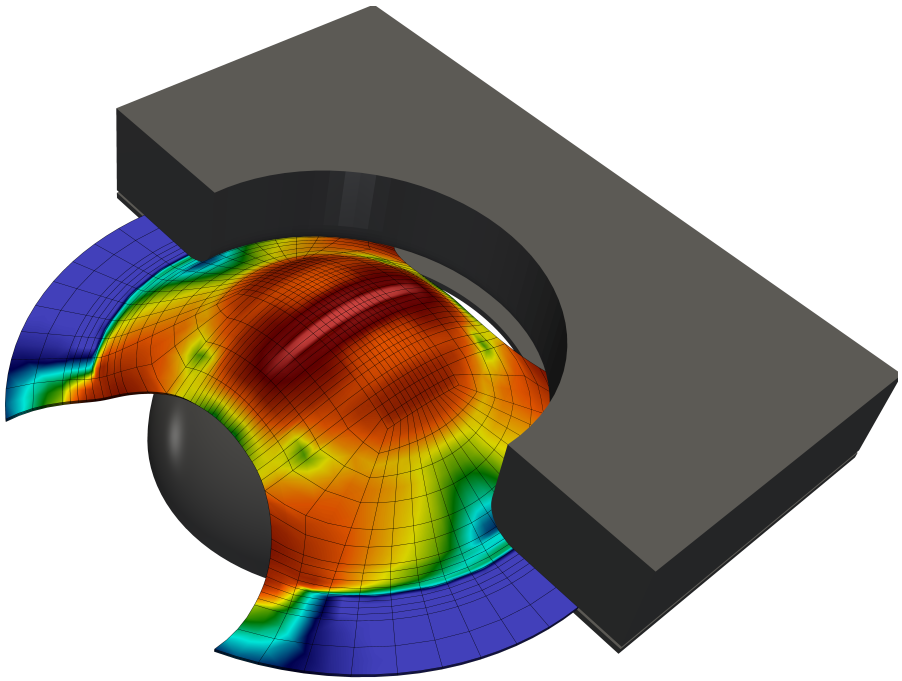
High-Performance Computing Two-Scale Finite Element Simulations of a Contact Problem Using Computational Homogenization

Virtual Forming Limit Curves for Dual-Phase Steel

Matthias Uran



High-Performance Computing Two-Scale Finite Element Simulations of a Contact Problem Using Computational Homogenization

Virtual Forming Limit Curves for Dual-Phase Steel

inaugural-dissertation

zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Matthias Uran
aus Moers

Köln, 2020

Berichterstatter: Prof. Dr. Axel Klawonn
(Universität zu Köln)

Prof. Dr. Oliver Rheinbach
(TU Bergakademie Freiberg)

Prof. Dr.-Ing. Jörg Schröder
(Universität Duisburg-Essen)

Tag der mündlichen Prüfung: 18. Mai 2020

Abstract

The appreciated macroscopic properties of dual-phase steels, which belong to the class of advanced high strength steels (AHSS), strongly depend on their microstructure. Therefore, accurate finite element (FE) simulations of a deformation process of such a steel require the incorporation of the microscopic heterogeneous structure. Usually, the typical length scale of the microscopic heterogeneities is a factor of up to 10^6 smaller compared to the macroscopic length scale. Therefore, a brute force finite element discretization incorporating the microstructure is not feasible since it results in exceedingly large problem sizes. Instead, the microstructure has to be incorporated by using computational homogenization.

In this thesis, we present a numerical two-scale approach of the Nakajima test for a dual-phase steel, which is a well known material test in the steel industry. It can be used to derive forming limit diagrams (FLDs), which allow experts to judge the maximum formability properties of a specific type of sheet metal in the considered thickness. For the simulations, we use our software package FE2TI, which is a highly scalable implementation of the well known FE^2 homogenization approach. The microstructure is represented by a representative volume element (RVE) and it is discretized separately from the macroscopic problem. Instead of considering an RVE representing a realistic microstructure, we use the concept of statistically similar RVEs (SSRVEs), which approximate the overall material properties. We discuss the incorporation of contact constraints using a penalty formulation as well as the considered sample sheet geometries and appropriate boundary conditions. In addition, we introduce a simple load step strategy and different opportunities for the choice of an initial value for a single load step by using an interpolation polynomial. Finally, we come up with computationally derived FLDs obtained from the implementation of two different evaluation strategies.

Although we use a computational homogenization strategy, the resulting problems on both scales can be quite large. The efficient solution of such large problems requires parallel strategies. Therefore, we consider the highly scalable nonlinear domain decomposition methods FETI-DP (Finite Element Tearing and Interconnecting - Dual-Primal) and BDDC (Balancing Domain Decomposition by Constraints). A nonlinear FETI-DP method has already been used for the parallel solution of large microscopic problems in a realistic simulation of a deformation process of dual-phase steel. For the first time, the BDDC approach is used for the parallel solution of the macroscopic problem in a simulation of the Nakajima test. We introduce a unified framework that combines all variants

of nonlinear FETI-DP and nonlinear BDDC. For the first time, we introduce a nonlinear FETI-DP variant that chooses suitable elimination sets by utilizing information from the nonlinear residual. Furthermore, we show weak scaling results for different nonlinear FETI-DP variants and several model problems.

Zusammenfassung

Die hervorragenden makroskopischen Eigenschaften von DP Stählen (DP=Dual-Phasen), die zu der Klasse von hochfesten Stählen (AHSS) gehören, hängen stark von der Mikrostruktur ab. Daher erfordern Finite-Elemente-Simulationen (FE=Finite Elemente) eines Umformprozesses eines DP Stahls die Einbeziehung der heterogenen Mikrostruktur. Für gewöhnlich ist die charakteristische Länge der mikroskopischen Heterogenitäten um einen Faktor von bis zu 10^6 kleiner im Vergleich zur charakteristischen Länge des Makroproblems. Dementsprechend ist eine direkte Finite-Elemente-Diskretisierung unter Berücksichtingung der mikroskopischen Heterogenitäten nicht zielführend, da wir zu große Probleme erhalten. Stattdessen muss die Mikrostruktur über einen Homogenisierungsansatz in die Simulation einfließen.

In dieser Arbeit präsentieren wir die Simulation des Nakajimatests für einen DP Stahl unter Verwendung zweiskaliger Finite-Elemente-Simulationen. Der Nakajimatest ist ein bekannter Materialtest, der in der Stahlindustrie dazu genutzt werden kann, Grenzformänderungsdiagramme (FLDs) zu erzeugen, anhand derer die Experten sofort die maximal zulässigen Umformungen für das betrachtete Blech mit entsprechender Blechdicke beurteilen können. Für die Simulationen nutzen wir unser Softwarepaket FE2TI, welches eine hochskalierbare Implementierung des bekannten FE^2-Homogenisierungsansatzes ist. Die Mikrostruktur wird durch sogenannte repräsentative Volumenelemente (RVEs) beschrieben und unabhängig vom Makroproblem diskretisiert. Anstelle einer realistischen Mikrostruktur nutzen wir das Konzept statistisch ähnlicher RVEs (SSRVEs), die das Materialverhalten gut approximieren. Wir diskutieren die Berücksichtiugung von Kontaktbedingungen unter der Verwendung einer Straftermformulierung sowie die betrachteten Probengeometrien und die passende Wahl von Randbedingungen. Zusätzlich stellen wir auch eine einfache Lastschrittstrategie und Möglichkeiten zu einer besseren Wahl eines Startwertes eines Lastschrittes vor. Für Letzteres verwenden wir Interpolationspolynome. Schlussendlich zeigen wir Grenzformänderungsdiagramme, die durch die Auswertung unserer Simulationsergebnisse mithilfe der implementierten Auswertungsverfahren generiert worden sind.

Obwohl wir einen Homogenisierungsansatz nutzen, können die Probleme auf Mikro- und Makroebene sehr groß werden. In diesem Fall erfordert die effiziente Lösung eines solchen Problems die Verwendung eines parallelen Lösers. Dazu betrachten wir unsere hochskalierbaren nichtlinearen Gebietszerlegungsverfahren FETI-DP (Finite Element Tearing and Interconnecting - Dual-Primal) und BDDC (Balancing Domain Decomposition by Constratins). Ein nicht-

lineares FETI-DP-Verfahren wurde bereits erfolgreich für das parallele Lösen großer Mikroprobleme in einer realistischen Simulation einer Stahlumformung genutzt. Für die Simulation des Nakajimatest wird in dieser Arbeit zum ersten Mal ein paralleler Löser - das BDDC-Verfahren - zur Parallelisierung des Makroproblems genutzt. Wir stellen ein einheitliches Framework vor, das alle nichtlinearen FETI-DP- und BDDC-Verfahren zusammenfasst. Zum ersten Mal wird ein problemabhängiges nichtlineares FETI-DP-Verfahren vorgestellt, das Informationen des nichtlinearen Residuums dazu nutzt, eine passende Eliminationsmenge zu bestimmen. Für die nichtlinearen FETI-DP-Verfahren zeigen wir Ergebnisse von schwachen Skalierbarkeitstests für verschiedene Modellprobleme.

Acknowledgements

First of all, I would like to thank my advisor Axel Klawonn for offering me the opportunity to work on some very interesting topics within his group during the last four years. I am especially grateful for the numerous and helpful discussions and his confidence in me, which I have felt throughout all the time. I would also like to thank Oliver Rheinbach for always bringing in new ideas in our discussions. Furthermore, his excellent implementations are the basis for the software used within this work. I am grateful to both for reviewing my thesis.

I am grateful to my current and former colleagues, namely Victor Grimm, Alexander Heinlein, Christian Hochmuth, Jascha Knepper, Martin Kühn, Patrick Radtke, and Janine Weber, who always had a good advice for me whenever I needed one. Special thanks go to Martin Lanser, whose excellent work in the first phase of the EXASTEEL project enabled most of the achievements of this work. In addition, he always took the time to calmly answer my countless questions. Without all of them, finishing this thesis would have been impossible.

Moreover, I would also like to thank the German Research Foundation (DFG) for the financial support through the Priority Programme 1648 "*Software for Exascale Computing*" (SPPEXA) and all my colleagues in the second phase of the EXASTEEL project, who are experts in their field and contributed to the completion of this thesis. Especially, I would like to thank Jörg Schröder, Daniel Balzani, Dominik Brands, Lisa Scheunemann, and Ashutosh Gandhi, who have implemented and provided the elasto-plastic material model and who were always helpful when I had questions regarding the experimental Nakajima test. I also want to thank Jörg Schrörder for reviewing my thesis. Furthermore, I am grateful to Stephan Köhler, who provided the discretizations of the representative volume elements that were used within this thesis.

I gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS and its predecessor JUQUEEN at Jülich Supercomputing Centre (JSC). I would also like to thank the Center for Computational Sciences and Simulation (CCSS) of the Universität of Duisburg-Essen for providing computing time on the supercomputer magnitUDE (DFG grants INST 20876/209-1 FUGG, INST 20876/243-1 FUGG) at the Zentrum für Informations- und Mediendienste (ZIM). In addition, the access to the computing cluster Taurus provided by the ZIH, Technical University of Dresden, is also gratefully acknowledged.

I would especially like to thank my sister Melanie and my parents, who have given me their full support on my way. Furthermore, I thank my parents for providing me with detailed information about steel, which goes far beyond the necessary knowledge of a mathematician.

Of course, I also want to thank my wife Frauke. With all her love and patience, she has given me the power to believe in myself, even in less successful times.

Special thanks go to my best buddy Chris, his wife Liesa, and their son Bruno for their understanding and all the time we spent together. I am pleased for every activity and especially my godchild Bruno always puts a smile on my face.

Finally, I would like to thank my sports team with whom I have spent hundreds of hours in cars and gyms without ever talking or thinking about mathematics.

To my parents. To Frauke.

Contents

List of Tables

List of Figures

1 Introduction

Towards a virtual laboratory for the computer-based derivation of specific properties of steel, this thesis considers the finite element simulation of contact-driven deformation processes of a dual-phase (DP) steel. DP steels combine strength and ductility and belong to the class of advanced high-strength steels. As a result, they are well-suited as components in safety-relevant parts of cars. In comparison to conventional steels, they provide the same resistance with less thickness. Consequently, the usage of DP steel in automobiles reduces weight and thus fuel consumption. The advantageous properties of DP steel result from the heterogeneous microstructure, which can be characterized by martensitic inclusions in a ferritic matrix material. Assuming that we have an appropriate chemical composition of the steel, the microstructure can be achieved by a precise and sophisticated heat treatment process during the rolling process, which is part of the production process of sheet metals.

Most of the results presented in this thesis have been developed within the second phase of the EXASTEEL project, which was part of the DFG (Deutsche Forschungsgemeinschaft) priority programme 1648 "*Software for Exascale Computing*" (2013-2019). In the last three years, six different groups were involved in this project, two each working in the fields of mathematics, engineering, and computer science, respectively. It was the goal of the EXASTEEL project to provide a realistic finite element simulation of the Nakajima test. This test is a material test used in the steel industry, in which a sheet metal is clamped between a blank holder and a die and is then deformed by a spherical rigid punch until it cracks.

Using different sample sheet geometries in the Nakajima test, certain evaluation strategies lead to a forming limit diagram (FLD), which provides permissible deformations for the considered thickness of the corresponding type of steel. It presents major and minor strains before plastic instabilities in a Cartesian coordinate system, where the minor strains are written on the x-axis. The different sample sheet geometries represent different strain paths, reaching from uniaxial to equi-biaxial tension. In addition, an interpolation polynomial is fitted to the different pairs of major and minor strains by using least squares.

The resulting polynomial is also part of the FLD and it is called forming limit curve (FLC).

To establish a virtual laboratory, it is mandatory that the simulation replicates the real experiment to the best possible extent. In particular, the simulation has to approximate the real deformation behavior quite well. Accordingly, we have to incorporate the microstructure into our simulations. For sure, the most simple idea to consider the microstructure would be to use a finite element discretization that resolves the microscopic heterogeneities. However, a sufficient discretization taking into account the microscopic heterogeneities requires finite elements that are a factor of up to 10^6 smaller compared to the macroscopic length scale. Accordingly, a brute force finite element discretization leads to problem sizes that cannot be solved, even on the largest current supercomputers.

Instead, we use a different strategy to incorporate the microstructure into our simulations. For this purpose, various homogenization approaches have been developed over the past decades, which allow the independent discretization of both levels; see, e.g., [30, 43–45, 48, 103, 104, 121, 124, 134, 135, 138]. As a result, the macroscopic problem, which is homogeneous from a macroscopic point of view, can be discretized using comparably large finite elements, which yields smaller macroscopic problems. For the microstructure, the concept of representative volume elements (RVEs) is used, which assumes that the overall microstructure can be represented by a small volume fraction. The RVEs are considered in selected macroscopic points, e.g., Gauß points, and, therefore, the microscopic problems can be solved in parallel.

Throughout this thesis, we use the FE^2 homogenization method; see, e.g., [43, 103,121,134,135,138]. The macroscopic problem is discretized by finite elements independent of the microstructure, and in each Gauß point, we solve a single boundary value problem on an RVE. The boundary values of the individual microscopic problems depend on the macroscopic deformation gradient in the corresponding points. Therefore, the microscopic problems are weakly coupled through the macroscopic problem. We exclusively use a phenomenological material law on the microscale and the macroscopic material law is replaced by using volumetric averages of stresses obtained from the corresponding microscopic problem. In addition, we also have to compute a consistent tangent modulus that takes into account the volumetric average of the tangent moduli of the corresponding microscopic problem.

Within this thesis, we show various results obtained from different production runs simulating the Nakajima test. Moreover, we present different FLDs and the

corresponding FLCs that are obtained by evaluating our simulation results. In addition, we discuss the numerical implementation as well as required software developments in our software package FE2TI.

Although we use the FE^2 method for considering the microscopic and the macroscopic levels separately, the resulting discretized problems can be very large on both scales, but especially on the microscale. As a result, they might be so large that they cannot be solved efficiently using a (sparse) direct solver. Accordingly, we have to use parallel solvers instead. So far, in the context of FE2TI, the FETI-DP method (Finite Element Tearing and Interconnecting - Dual Primal) [38, 39, 93, 94, 97, 98] has been used for the parallelization of the microscopic problem; see [74, 107]. The FETI-DP method belongs to the class of nonoverlapping domain decomposition methods. By introducing a further level of parallelization resulting from the application of an algebraic multigrid approach (AMG) to the FETI-DP coarse problem, a FE2TI-based simulation scaled to the complete JUQUEEN; see [78,79]. For further discussions regarding the combination of domain decomposition and computational homogenization, see, e.g., [19,57,143]. Moreover, some discretizations of the macroscopic sample sheet geometries within this thesis lead to macroscopic problems that cannot be solved efficiently by using a (sparse) direct solver. Accordingly, we have to use a parallel solution strategy. Within this thesis, the BDDC method (Balancing Domain Decomposition by Constraints) [26, 32, 111, 115, 116] is used for the first time for the parallel solution of the macroscopic problem in FE2TI-based simulations. The BDDC method also belongs to the class of nonoverlapping domain decomposition methods.

For the above-mentioned reasons, we are not only interested in the results of the production runs simulating the Nakajima test, but also focus on the development of highly scalable parallel solvers based on domain decomposition strategies. Since most mathematical formulations of realistic problems tend to be nonlinear, we consider solvers for nonlinear problems. To be more precise, we consider nonlinear versions of the nonoverlapping domain decomposition methods FETI-DP and BDDC. Both, the standard (linear) FETI-DP and BDDC methods are iterative substructuring methods that were derived from FETI-1 (Finite Element Tearing and Interconnecting) [12, 40–42, 96] and Balancing Neumann-Neumann [34, 96, 114], respectively; see also [150].

The conventional approach for the solution of nonlinear problems with domain decomposition strategies can be described as follows: Assuming that we have a discrete version of a nonlinear problem, this problem will be linearized first. Afterwards, any choice of a classical domain decomposition method can be

applied to the linearized system. This includes overlapping as well as nonoverlapping domain decomposition methods, such as FETI-1, FETI-DP, BDDC, and overlapping Schwarz [139, 150].

Typically, Newton's method is used for the linearization of the nonlinear problem due to its quadratic and thus optimal convergence in a neighborhood of the solution. However, other linearization techniques can also be used; see [18]. Using domain decomposition methods for the solution of nonlinear problems in the traditional way is referred to as Newton-Krylov-Domain-Decomposition, e.g, Newton-Krylov-FETI-DP or Newton-Krylov-BDDC. It can be characterized by linearization before decomposition.

Alternatively to the traditional approach, we can change the order of the operations linearization and decomposition, leading to nonlinear problems on the subdomains. The alternative approach has been applied for many different domain decomposition methods, e.g., nonlinear FETI-DP and BDDC, ASPIN (Additive Schwarz Preconditioned Inexact Newton) [21, 22, 50, 52–54, 65, 66, 100], MSPIN (Multiplicative Schwarz Preconditioned Inexact Newton) [112], RASPEN (Reduced Additive Schwarz Preconditioned Exact Newton) [33], nonlinear FETI-1 [126], and nonlinear Neumann-Neumann [13]. The nonlinear FETI-DP and BDDC methods are considered within this thesis and are based on nonlinear FETI-1. For a nonlinear FETI-DP method, scalability on Mira for almost 800 000 compute cores has been obtained for nonlinear elasticity; see [81]. Analogously to linear domain decomposition methods, we also have to incorporate a coarse problem to obtain good scalability for nonlinear domain decomposition approaches.

Within this thesis, we present a unified framework that combines all different variants of nonlinear FETI-DP and also nonlinear BDDC into a single framework. All these methods can be characterized by using a nonlinear right-preconditioner, which is associated with a (partial) nonlinear elimination of variables. It turns out that nonlinear FETI-DP methods usually improve the ratio of local work, global communication, and synchronization. Consequently, they reduce the time to solution compared to the traditional approach. In addition to the nonlinear FETI-DP variants presented in [77, 84, 85, 107], which have strictly defined elimination sets, we introduce a new variant of nonlinear FETI-DP that is tailored to the problem by choosing problem-dependent elimination sets.

We conclude the introduction by describing the structure of this thesis. In the first part (Chapters 2 to 4), we consider the simulation of the Nakajima test. In the following chapters, we focus on the nonlinear domain decomposition

methods nonlinear FETI-DP and nonlinear BDDC (Chapter 5), and we present results for nonlinear FETI-DP methods for different types of model problems (Chapter 6). In Chapter 7, we shortly summarize the presented results and give an outlook of possible future research activities.

The different parts are structured as follows. To deliver a necessary basis of knowledge, we first introduce the experimental test setup of the Nakajima test as well as two different evaluation strategies to create forming limit diagrams (Sections 2.1 and 2.2). If the reader is familiar with the Nakajima test, the first two sections can be skipped. In Section 2.3, we present the microscopic material law that is used throughout this thesis.. In the following, we describe the specifications of the sample sheet geometries and of the rigid tools that are used in the simulations (Section 2.4). In addition, we also discuss the choice of Dirichlet boundary conditions to approximate the real test setup as good as possible (see Section 2.5). We also present a strategy to reduce computational costs by utilizing the symmetric test setup of the Nakajima test; see Section 2.6. Beginning from Section 2.7, we show the criterion that is used to detect the appearance of a crack, and we introduce the numerical implementation of the considered evaluation strategies; see Sections 2.7 to 2.9. Finally, we give a detailed description of the computation of the major and minor strains that are essential for the derivation of FLDs.

The results regarding the different simulations of the Nakajima test are presented in Chapter 3. The FLDs and FLCs that we have obtained from applying the different evaluation strategies, which are introduced in Chapter 2, are of central importance. Furthermore, we discuss other aspects like the effects of different microstructures and penalty parameters, as well as the impact of the utilization of symmetry.

In Chapter 4, we give an overview of the software package FE2TI that has been developed during the last six years and is used for the simulations of the Nakajima test; see also [7]. We briefly discuss the computational homogenization approach FE^2 and present the most important software developments required for the simulation of the Nakjima test. Among others, it includes the incorporation of a contact algorithm using a penalty formulation.

In Chapter 5, we introduce the nonlinear variants of the nonoverlapping domain decomposition methods FETI-DP and BDDC, which can be used for the parallelization of both levels in the previously introduced FE2TI-based simulations. For a better understanding of the following subsections, we first introduce some basis notation and the classical linear FETI-DP approach. Subsequently, we present all nonlinear FETI-DP variants in a unified framework. In partic-

ular, we present a new variant that determines variable elimination sets with respect to the nonlinear residual; see also [50]. For the sake of completeness, we also present the traditional approaches NK-FETI-DP and NK-BDDC. With some further generalizations to the framework that covers all nonlinear FETI-DP variants, we can also incorporate the nonlinear BDDC approach into this framework.

In Chapter 6, we finally show numerical results for the FETI-DP approach for different nonlinear model problems regarding the p-Laplace equation and nonlinear elasticity.

We conclude this thesis by shortly summarizing the presented results and by discussing the possible future research activities.

2 The Nakajima Test

In the first part of this thesis, we deal with the simulation of the Nakajima test for the virtual determination of a forming limit diagram (FLD) and the corresponding forming limit curve (FLC). For a better understanding of the numerical implementation, a certain idea of the considered application is essential. Therefore, we first describe the test setup of the Nakajima test (see Figure 2.1) and look at the basic aspects necessary to determine an FLD; see Sections 2.1 and 2.2. Since we simulate the deformation of **dual-phase (DP) steel**, which belongs to the class of elasto-plastic materials, we use an elasto-plastic material model on the microscopic level, which is introduced in Section 2.3. Afterwards, we deal with the numerical implementation of the essential aspects; see all sections starting from Section 2.4. The corresponding numerical results are shown in the following chapter; see Chapter 3. Note that we mark all macroscopic quantities with an overline in the following to distinguish them from microscopic quantities. The contents of this chapter have been published in this or similar form by the author of this thesis and his coauthors in [87].

Climate protection is a dominant topic today. Almost every industry faces new challenges due to stricter CO_2 emission regulations all over the world, but especially in Europe. In addition, in the automotive industry, higher passenger safety norms are requested. In order to reduce the weight of a car without violating safety standards, lighter steel grades with higher toughness have to be used. These properties are fulfilled by steels of the class of DP steels, which belong to the class of advanced high-strength steels (AHSS) and combine strength and ductility. The advantageous macroscopic material properties of DP steels are achieved by a heterogeneous microstructure consisting of ferrite (first phase, soft) and martensite (second phase, hard); see, e.g., [9, 10]. In general, the microstructure results from the chemical composition of the steel and a complex heat treatment process, i.e., it strongly depends on the temperature and cooling before and during the rolling process, which is part of the production process of sheet metals; see, e.g., [9, 10, 145]. Let us note that DP steels can be produced by both, hot and cold rolling; see, e.g., [145]. For further details regarding the generation of a microstructure in general and for DP steel in particular, we refer to the literature mentioned in this paragraph.

For demonstration purposes, steel developers perform several material tests to proof the customers that their requirements to the produced sheet metal are fulfilled. A prominent member of material characterization is the **forming limit diagram (FLD)**, which contains major and minor strain values at failure initiation in a Cartesian coordinate system. In this context, material failure is already associated with the beginning of local necking in the direction of thickness and not only with crack formation [123, p. v]. The different major and minor strain points result from different strain paths reaching from uniaxial to equi-biaxial tension and represent the forming limits of a specific steel grade with a specific thickness. A regression curve of the determined major and minor strain values forms the transition between permissible and impermissible combinations of major and minor strains [123, p. v]. The regression curve is called **forming limit curve (FLC)** and gives the extent to which the material can be deformed by any combination of stretching and drawing without failing [123, p. v].

There are two standard procedures to derive an FLD, namely the Marciniak test and the **Nakajima test**, where the latter one is more common in practice [123]. In both tests, the specimen is clamped between a blank holder and a die and a punch is driven into the specimen from below or above until a crack occurs. The only difference between both tests is the shape of the punch. In the Nakajima test, we use a hemispherical punch while we use a flat circular punch in the Marciniak test; see [123, Sec. 4.3.3, 4.3.4] and also Figure 2.1. Note that it doens not matter whether the punch is driven into the specimen from below or above. For simplicity, we assume in the following that the punch is always located below the sample sheet. In both tests, the forming tool is allowed to move upwards with a speed of 1 to 2 mm per second.

For an accurate determination of the FLC with the Nakajima test, friction between the hemispherical punch and the specimen has to be avoided as much as possible [123, p. 2]. Therefore, different lubrication systems can be applied; see [123, Sec. 4.3.3.3]. A test is only valid if the tribological system is adjusted such that the crack occurs within a distance less than 15 % of the diameter of the punch away from the apex of the dome [123, Sec. 4.3.3.3].

Different pairs of major and minor strains are achieved by different shapes of sample geometries; see [123, Sec. 4.1.2] for a description of the recommended shapes of the sample sheet geometries and Section 2.4 for the shape of the sample sheets considered throughout this thesis. All geometries have in common that they have a central parallel shaft, which is perpendicular to the rolling direction for steel and parallel to the rolling direction for aluminium; see [123].

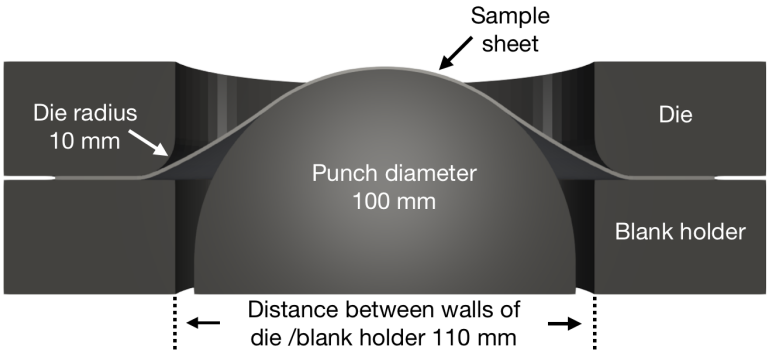


Figure 2.1: Cross section of the experimental test setup of the Nakajima test including the specifications of the punch, the blank holder, and the die that are used throughout this thesis.

For an FLC, at least five different shapes of sample geometries have to be used, and for every shape, three valid tests have to be carried out; see [123].

For evaluation purposes, the surface of a sample sheet is equipped with a regular grid or a stochastic pattern in experiments (see [123, Sec. 4.2]) and is recorded by one or more cameras during the deformation process.

There are at least two different strategies to get a pair of major and minor strains for the FLC, namely the **cross section method** [123] and a **method based on thinning rates** proposed by W. Volk and P. Hora [148]. Since the FLC gives information about material deformation without failing, we are interested in major and minor strains just before localized necking occurs. Both strategies are using the values evaluated from the images just before crack observation to get information about the values immediately before the beginning of localized necking.

The cross section method uses knowledge about the position of the crack and evaluates the last recorded image before crack along cross sections perpendicular to the crack. Then, from these values, the state immediately before material failure is interpolated; see Section 2.1.

In the method based on thinning rates, the last recorded image before the occurrence of localized necking is determined explicitly. This specific image is used to derive major and minor strains for the FLC; see Section 2.2.

2.1 Evaluation Based on the Cross Section Method

The cross section method is a standardized tool for the evaluation of the corresponding major and minor strain pairs for the FLD; see DIN EN ISO 12004-2:2008 [123, Ch. 5.2], where the presentations in this section are based on.

The evaluation based on the cross section method can be done in different ways. On one hand, the evaluation can be done on the cracked sample after removing it from the forming tool. On the other hand, the evaluation can also be performed using the last state before the crack becomes visible. In the following, we will consider the latter case. Note that the general idea of the cross section method is not affected by the different evaluation states, i.e., most parts of the following discussion also hold for the evaluation of the cracked sample sheet.

In the following, we assume that we have successfully performed the Nakajima test for a single sample sheet, i.e., the sample sheet cracked in the tolerated range and we have access to all recorded images of the sample sheet surface during the deformation process.

In the cross section method, the cross sections are considered to be as perpendicular as possible to the crack and have a length of at least 20 mm, including at least 10 grid points at both sides of the crack. For sample sheet geometries with a comparable small width of the parallel shaft, cross sections are supposed to be parallel to the shaft and in general, it is recommended that intersection lines correspond to the (virtual) grid orientation in the main strain directions [123]. The first cross section is placed such that its center is identical to the center of the crack and one or two cross sections are positioned above and below with a distance of about 2 mm.

The idea is to compute a pair of major and minor strains $\overline{\varepsilon}_1^{\mathrm{FLD}}$ and $\overline{\varepsilon}_2^{\mathrm{FLD}}$ for each cross section which represents the major and minor strains just before the beginning of plastic instability, i.e., before localized necking begins. For each grid point along an intersection line, we store the major and minor strain values as well as the corresponding x-position. Note that the position is computed from the arc length with reference to the first intersection point, which is associated with an x-position of 0 mm.

According to our assumption, we know the crack position as well as the first recorded image for which the crack becomes visible. For the determination of the cross sections, the first image with a visible crack is used. Afterwards, the cross sections are transferred to the previous image, i.e., the last image before the crack appears, which is subsequently used for the evaluation. Since we have no crack at this point, the crack position $\overline{p}^{\mathrm{cr}}$ is approximated for each cross

section separately. Therefore, we look for the maximum major strain value along the corresponding cross section and fit a second-order polynomial to the values within a range of 8 mm (4 mm on each side) or at least 5 grid points (2 at each side) by using least squares. The final crack position $\overline{p}^{\mathrm{cr}}$ is given by the location of the maximum value of the resulting second-order polynomial. Note that the definition of the crack position does not have to be performed if the cracked sample is evaluated.

The final major and minor strain values for the FLD are computed from the evaluation of two inverse second-order polynomials in the crack position. For the construction of the inverse second-order polynomials, we have to compute optimal fit windows on each side of the crack separately.

First of all, all grid points belonging to the necked area have to be determined. Therefore, for the major strains of each three consecutive grid points $i-1, i, i+1, i > 1$, a second-order polynomial g_i is constructed. For each g_i, the second derivative of g_i corresponds to the midpoint i of the three consecutive grid points. By construction, the grid point next to the crack has no associated second derivative. To smooth the second derivatives in a grid point i, the same is repeated using five consecutive grid points $i-2, \ldots, i+2,\ i > 2$. Subsequently, we generate a second second-order polynomial g_i^f by using the method of least squares. Note that the two grid points closest to the crack have no associated smoothed second derivative.

For all grid points with a maximum distance of 6 mm to the crack position $\overline{p}^{\mathrm{cr}}$, we have to find the local maxima of the second derivatives of g_i and g_i^f closest to $\overline{p}^{\mathrm{cr}}$. If the local maxima of the second derivatives of g_i and g_i^f differ by at least one grid point, the end of the necked area is defined by the position of grid point i with maximum second derivative g_i. Otherwise, the end of the necked area is defined as 3 mm away from $\overline{p}^{\mathrm{cr}}$. In case that there are less than four grid points with a maximum distance of 6 mm to the crack, the four grid points closest to $\overline{p}^{\mathrm{cr}}$ are considered.

To exclude values in the necked area from the interpolation, the end of the necked area is defined as inner boundary $\overline{b}_i$ of the fit window. Afterwards, we determine the optimal size $\overline{w}$ of the fit window by using knowledge about the major and minor strains $\overline{\varepsilon}_1^{i,l}$, $\overline{\varepsilon}_1^{i,r}$, $\overline{\varepsilon}_2^{i,l}$, and $\overline{\varepsilon}_2^{i,r}$ at the inner boundaries on both sides of the crack. The optimal window size w writes

$$\overline{w} = 10 \cdot \left(1 + \frac{0.5 \cdot (\overline{\varepsilon}_2^{i,l} + \overline{\varepsilon}_2^{i,r})}{0.5 \cdot (\overline{\varepsilon}_1^{i,l} + \overline{\varepsilon}_1^{i,r})}\right).$$

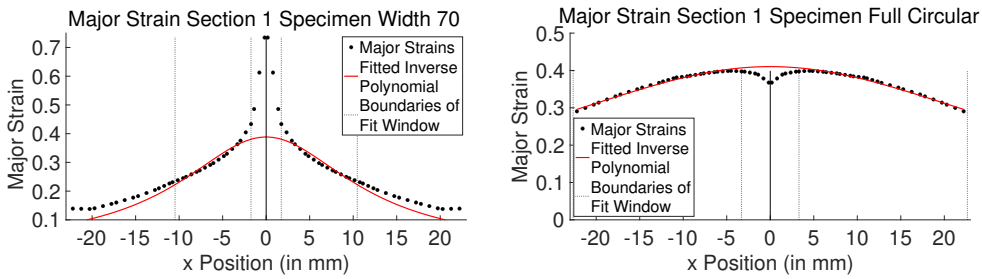


Figure 2.2: Fitted inverse secord-order polynomials to the major strain values along the first cross section. **Left:** Specimen with a width of the parallel shaft of 70 mm. **Right:** Full circular specimen.
This figure is accepted and soon to be published; see [87, Fig. 2]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE.

Finally, the outer boundary $\overline{b}_o = \overline{b}_i + \overline{w}$ is computed as the sum of the inner boundary $\overline{b}_i$ and the optimal window size $\overline{w}$.

After the fit windows on both sides of the crack are determined, we can start the interpolation process to recompute the major and minor strain values $\overline{\varepsilon}_1^{\mathrm{FLD}}$ and $\overline{\varepsilon}_2^{\mathrm{FLD}}$ for the FLD. At first, the major strain point $\overline{\varepsilon}_1^{\mathrm{FLD}}$ has to be determined. Therefore, an inverse second-order polynomial $\frac{1}{f(x)}$ has to be fitted to the major strain values $\overline{\varepsilon}_1$ within the fit windows on both sides of the crack by using the method of least squares. The evaluation of the inverse quadratic function at the crack position $\overline{p}^{\mathrm{cr}}$ yields the final major strain value $\overline{\varepsilon}_1^{\mathrm{FLD}} = \frac{1}{f(\overline{p}^{\mathrm{cr}})}$; see Figure 2.2 and Figure 2.7 (bottom). For the corresponding minor strain value $\overline{\varepsilon}_2^{\mathrm{FLD}}$, we again have to fit an inverse second-order polynomial $\frac{1}{h(x)}$ within the same fit windows as before by using least squares. Instead of fitting the inverse polynomial to the minor strain values $\overline{\varepsilon}_2$, the true thickness strain values $\overline{\varepsilon}_3 = -(\overline{\varepsilon}_1 + \overline{\varepsilon}_2)$ are used. Note that the true thickness strain is derived via the incompressibility equation (see [31, Eq. 3.103] and Equation (2.31)). Evaluation of the fitted inverse polynomial in the crack position $\overline{p}^{\mathrm{cr}}$ yields the true thickness strain $\overline{\varepsilon}_3^{\mathrm{FLD}} = \frac{1}{h(\overline{p}^{\mathrm{cr}})}$. Finally, the minor strain value $\overline{\varepsilon}_2^{\mathrm{FLD}}$ is computed from the incompressibility equation (see [31, Eq. 3.103] and Equation (2.31)) as

$$\overline{\varepsilon}_2^{\mathrm{FLD}} = -\left(\frac{1}{f(\overline{p}^{\mathrm{cr}})} + \frac{1}{h(\overline{p}^{\mathrm{cr}})}\right) = -\left(\overline{\varepsilon}_1^{\mathrm{FLD}} + \overline{\varepsilon}_3^{\mathrm{FLD}}\right).$$

For further details regarding the cross section method, we refer to [123].

Note that the derived values of major and minor strains $\overline{\varepsilon}_1^{\text{FLD}}$ and $\overline{\varepsilon}_2^{\text{FLD}}$ in general never exist during the deformation process. Consequently, these numbers somehow do not have a physical background [148]. Moreover, instead of fitting inverse second-order polynomials $\frac{1}{f(x)}$ and $\frac{1}{h(x)}$ to the major strain values $\overline{\varepsilon}_1$ as well as to the true thickness strain values $\overline{\varepsilon}_3$, we fit second-order polynomials $f(x)$ and $h(x)$ to the inverse of the corresponding values, i.e., $\frac{1}{\overline{\varepsilon}_1}$ and $\frac{1}{\overline{\varepsilon}_3}$.

2.2 Evaluation Method Based on Thinning Rates

Alternatively to the cross section method, another strategy to determine the corresponding major and minor strain values of the FLD is presented in [148], on which the discussion in this section is based on. In contrast to the cross section method, where the computed values usually do not exist during the deformation process, the method proposed in [148] determines the last image before the beginning of plastic instability by considering thinning rates. Subsequently, the major and minor strain values are evaluated from this specific image.

As before, let us assume that we have successfully performed the Nakajima test for a specific sample sheet and that we have access to all recorded images of the surface of the sample sheet during the deformation process. Let us further assume that the sample sheet surface is equipped with a regular grid. Accordingly, the recorded images of the sample sheet surface can be used to generate a finite element mesh using 4-node membrane elements. Subsequently, we can calculate thinning rates for each image and all finite elements.

The computation of thinning rates requires the deformation rate tensor

$$\overline{D} = 0.5 \cdot (\overline{L} + \overline{L}^T),$$

which can be computed from the velocity gradient tensor

$$\overline{L} = \dot{\overline{F}} \cdot \overline{F}^{-1},$$

where $\overline{F}$ is the 2D deformation gradient and $\dot{\overline{F}}$ its time derivative; see, e.g., [29, Sec. 3] for a detailed description of the tensors considered here. The thinning rate $\dot{\overline{\omega}} := \overline{d}_3 = -(\overline{d}_1 + \overline{d}_2)$ is now calculated by using the eigenvalues $\overline{d}_1$ and $\overline{d}_2$ of the deformation rate tensor $\overline{D}$ and the incompressibility equation (see Equation (2.31)). Note that the notation $\dot{\varepsilon}$ is used instead of $\dot{\overline{\omega}}$ in [148]. However, all macroscopic values are marked with an overline throughout this

thesis and $\bar{\varepsilon}$ is already reserved for the macroscopic strains. Therefore, we decided to use $\dot{\bar{\omega}}$ as notation for the thinning rates.

Before the beginning of plastic instability can be observed, a suitable database must first be determined. For a typical recording frequency of 10 images per second, the last 30 to 40 images before crack are recommended. Once the database is set, we have to compute for each image $k = 1, \dots, b$ in the database and for all elements $i = 1, \dots, g$ in the analysis area the thinning rates $\dot{\bar{\omega}}_i^k$. We introduce the ascending sorted set Γ^k of thinning rates

$$\overline{\Gamma}^k := \left\{ \dot{\bar{\omega}}_1^k, \dots, \dot{\bar{\omega}}_g^k \middle| \, \dot{\bar{\omega}}_{i-1}^k \leq \dot{\bar{\omega}}_i^k, i = 2, \dots, g \right\}, k = 1, \dots, b.$$

To define a set of elements that belong to the instability zone, we consider the second-last picture before the crack appears. We compute the representative maximum thinning rate $\dot{\bar{\omega}}_{\max}$, which is the arithmetic mean value of the five highest thinning rates, i.e.,

$$\dot{\bar{\omega}}_{\max} = \frac{1}{5} \sum_{i=g-4}^{g} \dot{\bar{\omega}}_i^{b-1}.$$

The set of elements in the instability zone is defined as

$$\mathcal{N} = \left\{ i \mid \dot{\bar{\omega}}_i^{b-1} \geq \alpha_{\mathcal{N}} \cdot \dot{\bar{\omega}}_{\max} \right\},$$

where $\alpha_{\mathcal{N}}$ can be chosen by the user. It is recommended to choose $\alpha_{\mathcal{N}}$ such that $\mathcal{N}$ contains 5 to 15 elements, depending on the grid size. For every image $k = 1, \dots, b-1$, the set

$$\mathcal{N}^k = \left\{ \dot{\bar{\omega}}_j^k \mid j \in \mathcal{N} \right\}$$

of thinning rates in the instability zone is stored and the representative thinning rate

$$\dot{\bar{\omega}}_{\text{rep}}^k = \frac{1}{|\mathcal{N}^k|} \sum_{i \in \mathcal{N}} \dot{\bar{\omega}}_i^k$$

is computed as the arithmetic mean value of thinning rates in $\mathcal{N}^k$.

Finally, the representative thinning rates $\dot{\bar{\omega}}_{\text{rep}}^k$, $k = 1, \dots, b$, are plotted over the number of images. This procedure yields a characteristic behavior for all specimens in the Nakajima test with a linear increase with a very small slope in the beginning, a linear increase with a high slope in the end, and a curved area in between; see Figure 2.3 and the corresponding figures in [148]. The small

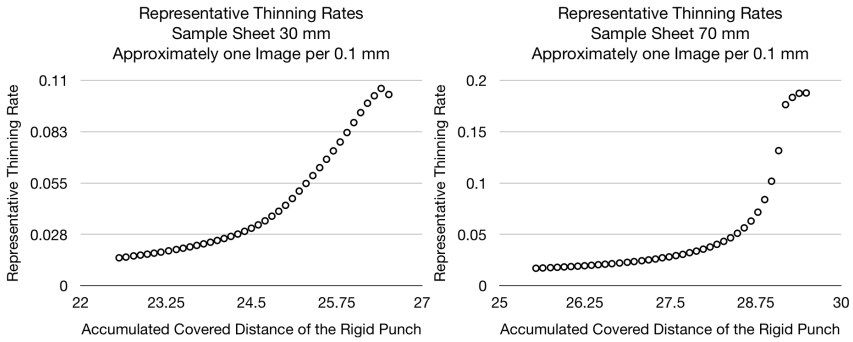


Figure 2.3: Representative thinning rates $\dot{\overline{\omega}}^k_{\text{rep}}$ for the last 4 mm before the crack appears with an approximative recording frequency of 10 images per millimeter and the rigid punch moves 1 mm per second. **Left:** Sample sheet geometry with a parallel shaft width of 30 mm. **Right:** Sample sheet geometry with a parallel shaft width of 70 mm.

increase of representative thinning rates $\dot{\overline{\omega}}^k_{\text{rep}}$ is associated with a stable deformation and the large increase is associated with localized necking. To define the point of beginning plastic instability, we have to fit linear functions to both linear parts of the diagram using least squares. Afterwards, we have to compute the cross point of both linear functions, which determines the beginning of plastic instability. In the end, the major and minor strains are computed as the arithmetic mean value of major and minor strains of all elements in the instability zone in the last picture before the beginning of plastic instability. For further details, we refer to [148].

2.3 Elasto-Plasticity

In the simulation of the Nakajima test, we deal with different sample sheets of a DP steel, which is a specific type of steel and therefore belongs to the class of elasto-plastic materials. Hence, accurate simulations require the usage of an elasto-plastic material model representing the different material behavior of the ferritic and martensitic phases on the microscopic level. Besides metals, also soils, rocks, and concrete belong to the class of plastic materials. Thus, plasticity is an important research topic in various fields, which is still up-to-date. In the last decades, also the numerical incorporation of crystal plasticity, which takes into account the polycrystalline structure of steels, is a large research topic; see, e.g., [129] and the references therein. For a deeper study of the theory of plasticity in the context of the continuum mechanical framework, we refer

to, e.g., [61,62]. For a deeper study of plasticity from a mathematical point of view, we refer to the references mentioned in [29], e.g., [55]. Following [29], the usage of a small strain plasticity model leads to inaccurate simulation results in the case of metal-forming operations. Therefore, we have to take into account the large strain plasticity model, which is introduced in, e.g., [29,35,137].

Plastic deformations of steel only take place when the material is subjected to further load after reaching a certain stress state. Before this state is reached, all deformations are completely elastic. For the determination of the beginning of plastic deformation, a yield criterion is required, which determines whether the deformation is completely elastic or contains plastic components based on the state of stress. The critical stress is called yield stress and can change during the deformation process. This change is defined by a hardening law. In addition, we also need a free energy potential and a flow rule. The latter is responsible for the evolution of the plastic quantities.

In the following, we derive all necessary functions for our elasto-plasticity material model following the discussion in [29]. The discussion in this section is also based on parts of [16,99]. We describe a material model that takes into account a rate-independent isotropic exponential-type hardening based on an associative von Mises yield criterion, which is also documented in, e.g., [16,99]; see also the references therein. Let us note that we consider strain-based hardening and that von Mises plasticity is also called J_2 plasticity, as is explained below.

When considering isotropic strain-based hardening, which is here the case, the hardening law usually depends only on a single scalar value representing the plastic strain; see [29, Sec. 6.6.2]. In the material model considered here, we use the effective von Mises plastic strain

$$\varepsilon^p = \int_0^t \sqrt{\frac{2}{3}\dot{\epsilon}^p : \dot{\epsilon}^p}\, \mathrm{d}t = \int_0^t \sqrt{\frac{2}{3}}||\dot{\epsilon}^p||\, \mathrm{d}t, \tag{2.1}$$

which is also called the von Mises equivalent plastic strain or, in short, equivalent plastic strain. Here, $\dot{\epsilon}^p$ is the rate form of the plastic component ϵ^p of the strain ϵ.

The free energy potential is a function depending on the overall strain and the hardening variable, i.e., it can be represented as a function $\psi(\epsilon, \varepsilon^p)$, where ϵ represents the strain. As it is assumed in the theory of small strains, we assume that the free energy potential can be additively split into an elastic and a plastic part, where the elastic part only depends on the elastic strain ϵ^e and

the plastic part depends on the hardening parameter, i.e., we can write

$$\psi(\epsilon, \varepsilon^p) = \psi^e(\epsilon^e) + \psi^p(\varepsilon^p), \tag{2.2}$$

where ϵ^e is the elastic part of the overall strain. To formulate the yield function, the thermodynamical force

$$\beta := \beta(\varepsilon^p) = \frac{\partial \psi}{\partial \varepsilon^p} = \frac{\partial \psi^p}{\partial \varepsilon^p} \tag{2.3}$$

is introduced, which is conjugate to the hardening parameter ε^p.

The yield function is a scalar function depending on a stress tensor, which is the Kirchhoff stress tensor τ (see [29, Sec. 3.3.6]), and the thermodynamical force β. Accordingly, the yield function can be written as a function $\Phi(\tau, \beta)$. Note that β defines the yield stress, i.e., the critical stress at which plastic yielding begins. For all stress states which fulfill $\Phi(\tau, \beta) < 0$, the deformation is fully elastic. Therefore, we can introduce the elastic domain of stress states as

$$\mathcal{E}_\Phi = \left\{ \tau \;\middle|\; \Phi(\tau, \beta) < 0 \right\}. \tag{2.4}$$

The boundary of the elastic domain $\mathcal{E}_\Phi$ is defined by

$$\mathcal{Y}_\Phi = \left\{ \tau \;\middle|\; \Phi(\tau, \beta) = 0 \right\}, \tag{2.5}$$

which is also called yield surface. It defines the set of Kirchhoff stresses τ at which plastic deformations may occur. Let us note that the set of admissible Kirchhoff stresses is defined by

$$\mathcal{A}_\Phi = \left\{ \tau \;\middle|\; \Phi(\tau, \beta) \leq 0 \right\}. \tag{2.6}$$

As mentioned before, we consider a von Mises yield criterion, where the onset of plastic flow depends on the J_2 stress deviator invariant. This is why the von Mises plasticity is also called J_2 plasticity. The deviator of the Kirchhoff stress τ is defined as

$$\mathrm{dev}(\tau) = \tau - \frac{1}{3}\mathrm{tr}(\tau) I$$

and J_2 is defined as the negative second invariant, i.e.,

$$\mathrm{J}_2 := \mathrm{J}_2(\mathrm{dev}(\tau)) = -\mathrm{I}_2(\mathrm{dev}(\tau)) = \frac{1}{2}\mathrm{tr}(\mathrm{dev}(\tau)^2) = \frac{1}{2}||\mathrm{dev}(\tau)||^2.$$

The von Mises yield criterion can be written as

$$\Phi(\tau, \beta) = \sqrt{3 \cdot \mathrm{J}_2(\mathrm{dev}(\tau))} - \beta = \sqrt{\frac{3}{2}} \cdot ||\mathrm{dev}(\tau)|| - \beta. \tag{2.7}$$

In associative plasticity models, the yield criterion is equivalent to the flow rule.

The evolution of the plastic variables has to be derived from a thermodynamical law. Therefore, we first have to introduce some relevant tensors. Let us note that we do not go too much into detail, and we refer to the literature, e.g., [29], for a more detailed discussion.

In large strain plasticity, the deformation gradient F is multiplicatively split into the elastic deformation gradient F^e and the plastic deformation F^p, i.e., we can write

$$F = F^e \cdot F^p, \tag{2.8}$$

which was introduced in [109, 110]; see also [105]. Here, the elastic deformation gradient belongs to the stress-driven deformations and the plastic deformation gradient $F^p = (F^e)^{-1}F$ belongs to a stress-free intermediate configuration. Locally, the intermediate configuration can be interpreted as the result of elastic unloading (multiplication with the inverse of F^e) of the overall deformation F. The multiplicative split of the deformation gradient also finds a solid physical justification in the slip theory of crystals; see [29, Rem. 4.1]. Analogously to the deformation gradient F, we can perform the polar decomposition of the elastic and plastic deformation tensors resulting in

$$\begin{aligned} F^e &= R^e U^e = V^e R^e \\ F^p &= R^p U^p = V^p R^p, \end{aligned} \tag{2.9}$$

where the tensors U^j, V^j and R^j, $j \in \{e, p\}$, are denoted as elastic/plastic right stretch tensor, left stretch tensor, and rotation tensor, respectively; see [29, Sec. 2.2.9]. Similar to the split of the deformation gradient into elastic and plastic parts, we can also define elastic and plastic parts of the right and left Cauchy-Green strain tensors $C = F^T F$ and $B = FF^T$; see [29] for details about the different strain tensors. Defining

$$\begin{aligned} C^e &= (F^e)^T F^e, & C^p &= (F^p)^T F, \\ B^e &= F^e (F^e)^T, & B^p &= F^p (F^p)^T, \end{aligned}$$

the multiplicative decomposition of the deformation gradient F (see Equation (2.8)) yields

$$B = F^e F^p (F^p)^T (F^e)^T = F^e B^p (F^e)^T \quad \text{and}$$
$$C = (F^p)^T (F^e)^T F^e F^p = (F^p)^T C^e F^p.$$

Inserting Equation (2.8) into the velocity gradient $L = \dot{F}F^{-1}$, we obtain

$$\begin{aligned} L = \dot{F}F^{-1} &= \dot{F}^e F^p F^{-1} + F^e \dot{F}^p F^{-1} \\ &= \dot{F}^e F^p (F^p)^{-1} (F^e)^{-1} + F^e \dot{F}^p (F^p)^{-1} (F^e)^{-1} \\ &= \dot{F}^e (F^e)^{-1} + F^e \dot{F}^p (F^p)^{-1} (F^e)^{-1} \\ &:= L^e + F^e L^p (F^e)^{-1}. \end{aligned}$$

Analogously to the stretch tensor (also called deformation rate tensor) $D = \frac{1}{2}(L + L^T) = \mathrm{sym}(L)$, we can also define the plastic stretch tensor

$$D^p = \mathrm{sym}(L^p) = \mathrm{sym}(\dot{F}^p (F^p)^{-1}). \tag{2.10}$$

Note that any tensor T can be decomposed into its symmetric and skew-symmetric part, i.e., we can write

$$T = \frac{1}{2}(T + T^T) + \frac{1}{2}(T - T^T) := \mathrm{sym}(T) + \mathrm{skew}(T);$$

see [29, Sec. 2.2.1]. We also define $W^p := \mathrm{skew}(L^p)$, which is assumed to be zero throughout this thesis. This assumption is compatible with plastic isotropy; see [29]. Again, for a more detailed discussion, we refer to, e.g., [29].

Assuming that D^p has the orthonormal eigenvectors v_i and corresponding eigenvalues d_i^p, $i = 1, \ldots, 3$, we can write D^p in its spectral decomposition as

$$D^p = \sum_{i=1}^{3} d_i^p v_i \otimes v_i.$$

The plastic stretch tensor measures the plastic strain along the orthogonal directions defined by the orthonormal eigenvectors v_i, $i = 1, \ldots, 3$. To consider the spatial configuration, we have to consider the plastic stretch tensor in the deformed or spatial configuration, i.e., after applying the elastic rotation R^e (see Equation (2.9)) to D^p (see Equation (2.10)). We obtain

$$\widetilde{D}^p = R^e D^p (R^e)^T = R^e \mathrm{sym}(L^p)(R^e)^T.$$

Note that D^p and $\widetilde{D}^p$ have the same eigenvalues and the corresponding eigenvectors of $\widetilde{D}^p$ are $R^e \cdot v_i$, $i = 1, \ldots, 3$, where v_i are the orthonormal eigenvectors of D^p. For further details, we refer to [29].

Throughout this thesis, we consider logarithmic strain measures using the left Cauchy-Green tensor B, i.e., we obtain

$$\epsilon^e = \frac{1}{2}\ln(B^e). \tag{2.11}$$

Recalling that the elastic part of the free energy only depends on the elastic strain (see Equation (2.2)), we obtain

$$\psi^e(\epsilon^e) = \frac{\lambda}{2}(\ln(\lambda_1\lambda_2\lambda_3))^2 + \mu((\ln\lambda_1)^2 + (\ln\lambda_2)^2 + (\ln\lambda_3)^2), \tag{2.12}$$

where $(\lambda_i^e)^2$, $i = 1, \ldots, 3$ are the eigenvalues of the elastic part of the left (or right) Cauchy-Green tensor; see [29, Eq. (13.58)].

The constitutive law for the Kirchhoff stress is now derived from the Clausius-Duhem inequality (see [16, 29]). Neglecting all thermal aspects, we obtain

$$\tau : D - \overline{\rho}\dot{\psi} \geq 0, \tag{2.13}$$

where $\overline{\rho}$ is the mass density of the reference configuration and $\dot{\psi}$ is the time derivative of the free energy. From the additive decomposition of the free energy (see Equation (2.2)), we obtain

$$\dot{\psi}(\epsilon^e, \varepsilon^p) = \frac{\partial\psi^e}{\partial\epsilon^e} : \dot{\epsilon}^e + \frac{\partial\psi^p}{\partial\varepsilon^p}\dot{\varepsilon}^p. \tag{2.14}$$

Inserting Equation (2.11) into the latter equation, we obtain

$$\dot{\psi}(\epsilon^e, \varepsilon^p) = \frac{\partial\psi^e}{\partial\epsilon^e} : (D - \widetilde{D}^p) + \frac{1}{\overline{\rho}}\frac{\partial\psi^p}{\partial\varepsilon^p}\dot{\varepsilon}^p. \tag{2.15}$$

Again, inserting Equation (2.15) into Equation (2.13), we obtain

$$\left(\tau - \overline{\rho}\frac{\partial\psi^e}{\partial\epsilon^e}\right) : D + \overline{\rho}\frac{\partial\psi^e}{\partial\epsilon^e} : \widetilde{D}^p - \frac{\partial\psi^p}{\partial\varepsilon^p}\dot{\varepsilon}^p \geq 0,$$

which has to be fulfilled for all tensors D, i.e., the constitutive equation

$$\tau = \overline{\rho}\frac{\partial\psi^e}{\partial\epsilon^e} \tag{2.16}$$

has to hold; see [29]. Finally, Equation (2.13) reduces to

$$\tau : \widetilde{D}^p - \frac{\partial \psi^p}{\partial \varepsilon^p} \dot{\varepsilon}^p \geq 0. \tag{2.17}$$

Let us note that in [16, 99] the constitutive equation $\tau = 2\frac{\partial \psi^e}{\partial B^e} B^e$ is derived, which is identical to our result since we have chosen $\epsilon^e = \frac{1}{2}\ln(B^e)$ and, therefore, we obtain

$$\frac{\partial \psi}{\partial \epsilon^e} = \frac{\partial \psi}{\partial \frac{1}{2}\ln(B^e)} = 2\frac{\partial \psi}{\partial B^e}\frac{\partial B^e}{\partial \ln(B^e)} = 2\frac{\partial \psi}{\partial B^e} B^e.$$

The missing factor $\overline{\rho}$ results from the slightly different definition of $\dot{\psi}$ in [16], where the factor is already introduced in $\dot{\psi}$.

From the principle of maximum dissipation, we obtain that the current state of τ and β has to maximize Equation (2.17). Therefore, we have to solve an optimization problem under the additional constraint that the stress state belongs to the set of admissible stresses. Instead of maximizing Equation (2.17), we consider the equivalent minimization of

$$-\tau : \widetilde{D}^p + \frac{\partial \psi^p}{\partial \varepsilon^p} \dot{\varepsilon}^p \leq 0.$$

Therefore, we introduce the Lagrange multiplier $\dot{\gamma}$ and obtain the Lagrange functional

$$-\tau : \widetilde{D}^p + \frac{\partial \psi^p}{\partial \varepsilon^p} \dot{\varepsilon}^p + \dot{\gamma}\Phi(\tau, \beta) \leq 0. \tag{2.18}$$

Building the partial derivatives of the Lagrange functional with respect to β and τ yield the evolution equation of the hardening variable ε^p and the flow rule, respectively. For the evolution of the hardening parameter, we obtain

$$\dot{\varepsilon}^p = -\dot{\gamma}\frac{\partial \Phi(\tau, \beta)}{\partial \beta}, \tag{2.19}$$

where we have used the relation $\beta = \frac{\partial \psi^p}{\partial \varepsilon^p}$ (see Equation (2.3)). The plastic flow rule is given by the constitutive equation

$$\widetilde{D}^p = \dot{\gamma}\frac{\partial \Phi}{\partial \tau}. \tag{2.20}$$

The partial derivative of Equation (2.18) with respect to the Lagrange multiplier $\dot{\gamma}$ yields $\Phi(\tau, \beta) = 0$. As usual, Equations (2.19) and (2.20) and $\Phi(\tau, \beta) = 0$ are solved iteratively and the fulfillment of the Lagrange function is

ensured by the well-known Karush-Kuhn-Tucker (KKT) optimality conditions (see, e.g., [11, 113, 122])

$$\Phi(\tau, \beta) \leq 0, \quad \dot{\gamma} \geq 0, \quad \Phi(\tau, \beta) \cdot \dot{\gamma} = 0. \tag{2.21}$$

In terms of loading and unloading of sheet metals, we can interpret the KKT conditions as follows. In the case of elastic loading, the yield criterion is fulfilled, i.e., $\Phi < 0$ holds; see Equation (2.4). In this case, the third equation of the KKT condition (2.21) yields $\dot{\gamma} = 0$, i.e., the plastic variable stays unchanged since $\dot{\varepsilon}^p = 0$ is obtained from Equation (2.19). Alternatively, we have to consider the case when we have a stress state $\tau \in \mathcal{Y}_\Phi$, i.e., $\Phi = 0$ holds; see Equation (2.5). From the definition of the admissible set of stress states (see Equation (2.6)), it follows that the yield criterion can not rise in the next step, i.e., it stays constant ($\dot{\Phi} = 0$), or it decreases ($\dot{\Phi} < 0$). The latter case is associated with elastic unloading since we obtain $\dot{\gamma} = 0$ as before. If we have $\dot{\Phi} = 0$, two different states for the Lagrange multiplier are possible. First, the Lagrange multiplier can also be zero, i.e., $\dot{\gamma} = 0$, which is associated with a neutral loading. Otherwise, $\dot{\gamma} > 0$ holds, which represents plastic loading. Pairwise combination of the values of the individual cases yields the consistency equation

$$\dot{\gamma}\dot{\Phi}(\tau, \beta) = 0;$$

see [16]. Let us note that it is also possible to define the flow rule (see Equation (2.20)) using the directional derivative of the elastic left Cauchy-Green tensor B^e, which is also called the LIE derivative; see [119]. From [16], we obtain the representation

$$\mathcal{L}(B^e) = F\left(\frac{\partial}{\partial t}(F^{-1}B^eF^{-T})\right)F^T = F(\dot{C}^p)^{-1}F^T, \tag{2.22}$$

where the last equation follows from

$$F^{-1}B^eF^{-T} = (F^p)^{-1}(F^e)^{-1}F^e(F^e)^T(F^e)^{-T}(F^p)^{-T} = (F^p)^{-1}(F^p)^{-T} = (C^p)^{-1}.$$

In [29], the LIE derivative is represented as

$$\mathcal{L}(B^e) = \dot{B}^e - LB^e - B^eL^T,$$

which can be used to reformulate the flow rule (see Equation (2.20)) to

$$-\frac{1}{2}\mathcal{L}(B^e) \cdot (B^e)^{-1} = \dot{\gamma}\frac{\partial \Phi}{\partial \tau};$$

see also [16, 99]. Inserting Equation (2.22) into the last equation and using $F^e = F(F^p)^{-1}$ as well as $(\dot{C}^p)^{-1}C^p = -(C^p)^{-1}\dot{C}^p$, we obtain

$$\dot{C}^p = 2\dot{\gamma}C^p(F^{-1}\frac{\partial\Phi}{\partial\tau}F)$$

as an alternative flow rule in terms of the plastic part of the right Cauchy-Green tensor; see again [16, 99].

To complete the formulation of the material model, we have to summarize the important results from this section and also have to formulate a hardening law, where we use an exponential hardening rule. For the yield criterion, we do not use the derived function $\widetilde{\Phi}(\tau,\beta) = \sqrt{\frac{3}{2}}||\mathrm{dev}(\tau)|| - \beta$ (see Equation (2.7)) but the equivalent representation

$$\Phi(\tau,\beta) = \sqrt{\frac{2}{3}}\widetilde{\Phi}(\tau,\beta) = ||\mathrm{dev}(\tau)|| - \sqrt{\frac{2}{3}}\beta; \tag{2.23}$$

see also [16]. Inserting the yield criterion (see Equation (2.23)) into the evolution equation of the plastic strain (see Equation (2.19)), the evolution of the equivalent plastic strain writes

$$\dot{\varepsilon}^p = \frac{\partial\Phi}{\partial\beta}\dot{\gamma} = \sqrt{\frac{2}{3}}\dot{\gamma}. \tag{2.24}$$

For the variable β, which is conjugate to the equivalent plastic strain, we write

$$\beta = y_\infty + (y_0 - y_\infty)\exp(-\eta\varepsilon^p) + h\varepsilon^p, \tag{2.25}$$

where y_0 is the initial yield strength and y_∞ as well as η are material parameters representing the exponential hardening behavior. The variable h defines the superimposed linear hardening; see [16]. Combining the definition of β in Equation (2.25) and the relation $\beta = \frac{\partial\psi^p}{\partial\varepsilon^p}$ (see Equation (2.3)), we obtain the plastic part of the free energy $\psi(\epsilon,\varepsilon^p)$ as

$$\psi^p(\varepsilon^p) = y_\infty\varepsilon^p - \frac{1}{\eta}(y_0 - y_\infty)\exp(-\eta\varepsilon^p) + \frac{1}{2}h(\varepsilon^p)^2. \tag{2.26}$$

All necessary quantities for the derivation of the material model are summarized in Table 2.1.

Implicit Integration Scheme Since elasto-plastic materials are path-dependent (see, e.g., [29]), i.e., the stress tensor does not only depend on the current strain but also on the history of the plastic strain, we have to

kinematics	$F = F^e F^p$, $B^e = F^e (F^e)^T$, $\epsilon^e = \frac{1}{2}\ln(B^e)$	
strain energy	$\psi = \psi^e(\epsilon^e) + \psi^p(\varepsilon^p)$	see (2.2)
elastic part	$\psi^e = \frac{\lambda}{2}(b_1+b_2+b_3)^2 + \mu(b_1^2+b_2^2+b_3^2)$ $b_i = \ln(\lambda_i^e)$, λ_i^e eigenvalues of B^e, $i = 1,\dots,3$.	see (2.12)
plastic part	$\psi^p = y_\infty \varepsilon^p - \frac{1}{\eta}(y_0 - y_\infty)\exp(-\eta\varepsilon^p) + \frac{1}{2}h(\varepsilon^p)^2$	see (2.26)
stresses	$\tau = \overline{\rho}\frac{\partial \psi^e}{\partial \epsilon^e} = 2\frac{\partial \psi^e}{\partial B^e}B^e = \sum_{i=1}^3 \tau_i \cdot v_i \otimes v_i$, $\tau_i = \frac{\partial \psi^e}{\partial b_i}$, v_i orthonormal eigenvectors of B^e	
conj. internal	$\beta = y_\infty + (y_0 - y_\infty)\exp(-\eta\varepsilon^p) + h\varepsilon^p$	see (2.25)
yield criterion	$\Phi = \|\|\mathrm{dev}(\tau)\|\| - \sqrt{\frac{2}{3}}\beta$	see (2.23)
flow rule	$\widetilde{D}^p = \dot{\gamma}\frac{\partial \Phi}{\partial \tau}$	see (2.20)
evolution of plastic variable	$\dot{\varepsilon}^p = \sqrt{\frac{2}{3}} \cdot \dot{\gamma}$	see (2.24)
KKT conditions	$\Phi \leq 0, \quad \dot{\gamma} \geq 0, \quad \Phi\dot{\gamma} = 0$	see (2.21)

Table 2.1: Summary of the finite J_2 elasto-plasticity material model; see also [16].

use a special numerical treatment for the integration of the flow rule. In [29], it is stated that the stress tensor is obtained from solving a constitutive initial boundary value problem. Therefore, the integration of the flow rule is necessary, which is a rate constitutive equation and is therefore integrated using a time discretization. In our case, we will use an implicit Euler scheme for the discretization and the integration is done using an exponential update algorithm. Following [99], it was originally proposed by Weber and Anand in [149]. Using a von Mises flow rule, which is considered throughout this thesis, plastic incompressibility is preserved throughout the update procedure, which is necessary for deformation processes of metals; see [16, 29, 99] and the references therein.

We are interested in the evolution of the plastic variables from time t_n to time t_{n+1}. All quantities belonging to time t_n are already known and are denoted with an index $(\cdot)_n$. Analogously, all quantities belonging to time t_{n+1} are denoted with an index $(\cdot)_{n+1}$. Note that the relevant quantities at time t_{n+1} are unknown and only the deformation gradient F_{n+1} is known since we consider a deformation driven formulation. The admissible stress state τ_{n+1} and the equivalent plastic strain ε^p_{n+1} are obtained by a return-mapping algorithm to the time interval $[t_n, t_{n+1}]$. Within this algorithm, it is assumed that the

deformation from time t_n to time t_{n+1} is completely elastic, i.e., the plastic quantities do not change. Therefore, we introduce a trial state in time t_{n+1} and set

$$(C^p_{n+1})^{\text{trial}} = C^p_n, \qquad (\varepsilon^p_{n+1})^{\text{trial}} = \varepsilon^p_n.$$

These quantities are used to compute $\tau^{\text{trial}}_{n+1}$ and $\beta^{\text{trial}}_{n+1}$, which are then used to check whether the flow criterion $\Phi^{\text{trial}}_{n+1}(\tau^{\text{trial}}_{n+1}, \beta^{\text{trial}}_{n+1}) \leq 0$ is fulfilled. If the condition holds, the initial assumption of a fully elastic deformation is true and we set

$$\begin{aligned} C^p_{n+1} &= (C^p_{n+1})^{\text{trial}} = C^p_n, & \tau_{n+1} &= \tau^{\text{trial}}_{n+1}, \\ \varepsilon^p_{n+1} &= (\varepsilon^p_{n+1})^{\text{trial}} = \varepsilon^p_n, & \beta_{n+1} &= \beta^{\text{trial}}_{n+1}. \end{aligned}$$

Otherwise, the trial stresses have to be projected onto the yield surface, i.e., on the boundary of the domain of admissible stresses (see Equation (2.5)). Due to the exponential (nonlinear) hardening law, the projection cannot be derived in a closed form but has to be computed by a Newton iteration. For further details regarding the algorithmic treatment of the Newton iteration, we refer to [99]. In case of isotropic hardening, which is considered throughout this thesis, the return-mapping algorithm is also called radial return algorithm. Following [29], it was the first type of return-mapping algorithm.

2.4 Sample Sheet Geometries and Specifications of Rigid Tools

So far, we have introduced the test setup of the Nakajima test as well as two different evaluation strategies for pairs of major and minor strain values for the FLD. Furthermore, we have discussed the implemented material model on the microscopic level. In the following, we will focus on the numerical implementation of the application introduced above.

The sample sheet geometries we consider within this thesis fit to the normed range in DIN EN ISO 12004-2:2008 [123]. Furthermore, they were partly used in another PhD thesis by David Jocham; see [68]. Most of the sample sheets have a central parallel shaft, where the length of the shaft is 25 mm, and its width varies from 30 mm to 129 mm; see Figure 2.4 for some examples. These geometries are expected to yield a more uniform distribution of the experimental forming limit points than rectangular sample sheets and are therefore recommended; see [123]. Additionally, we consider a fully circular specimen; see

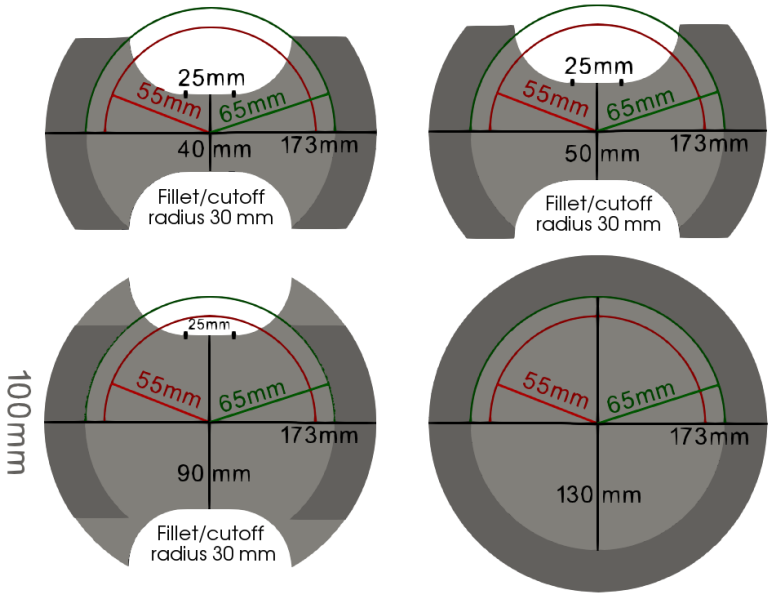


Figure 2.4: Specifications of sample sheets with different parallel shaft widths and of the full circular sample sheet. The dark gray area is assumed to be fully clamped, i.e., we apply Dirichlet boundary conditions; see Section 2.5. The red (inner) circle represents the inner wall of the die and the green (outer) circle represents the beginning of the clamped part between the blank holder and the die; see also [87, Fig. 1 (right)] for a similar figure.

again Figure 2.4. Throughout this thesis, we exclusively handle sample sheets with a thickness of 1 mm.

For all specimens, the material is assumed to be completely clamped by a bead, which has a radius of 86.5 mm. Therefore, we only consider material points $\overline{p} = [\overline{p}_x, \overline{p}_y, \overline{p}_z]$ which fulfill $\sqrt{(\overline{p}_x - \overline{c}_x)^2 + (\overline{p}_y - \overline{c}_y)^2} \leq 86.5$ mm, where $\overline{c} = [\overline{c}_x, \overline{c}_y, \overline{c}_z]$ is the center of the sample sheet; see Figure 2.4. Throughout this thesis, the center $\overline{c}^b = [\overline{c}_x, \overline{c}_y, \overline{c}_z^b]$ of the bottom surface of the discretized sample sheet is always placed in the origin of the coordinate system.

As for the considered sample sheets, also the specifications of the rigid tools are within the normed range from [123]. The hemispherical punch has a radius of 50 mm. The blank holder is a square plate of 173 mm $\times$ 173 mm that has a circular whole in the middle with a radius of 55 mm; see the red (inner) circle in Figure 2.4. The inner wall of the die also has a radius of 55 mm, i.e., it is placed with a distance of 5 mm to the rigid punch; see, again, the inner (red) circle in Figure 2.4. The die radius is chosen to be 10 mm (see Figure 2.1),

i.e., all material points $\overline{p}$ with $\sqrt{(\overline{p}_x - \overline{c}_x)^2 + (\overline{p}_y - \overline{c}_y)^2} = \sqrt{\overline{p}_x^2 + \overline{p}_y^2} \geq 65$ are possibly clamped between the blank holder and die.

Following the discussion on appropriate boundary conditions in Section 2.5, we only consider finite element nodes $\overline{p}$ which fulfill $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} \leq 65$ for all specimens with a shaft width of less than 90 mm . The same holds for the completely circular sample sheet. Only for sample sheets with larger shaft widths, we consider finite element nodes that fulfill $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} \leq 86.5$.

2.5 Choosing Appropriate Boundary Conditions

To obtain accurate solutions for our finite element simulations of the Nakajima test, we have to approximate the real test conditions as good as possible. For this reason, we have to incorporate the blank holder and the die into the simulation process. Otherwise, the deformation behavior of the sheet metal would be different, which might lead to a wrong failure zone. Furthermore, the forming tool has to cover a larger distance until failure occurs. The consideration of the blank holder and the die in the simulation of the Nakajima test becomes even more relevant if we consider friction. In this case, the frictional sliding of the sheet metal along with the blank holder and the die has an impact on the deformation behavior. Therefore, friction affects the time at which the crack occurs.

Since the blank holder and the die force the metal sheet into a specific shape by contact, the incorporation of these tools raises the number of possible contact points in our simulations. Accordingly, we have to add additional contact terms to more finite element nodes, which enlarges the assembly time on one hand and the complexity of the problem on the other hand. We have come up with various strategies for taking the blank holder and the die into account without increasing the number of contact points.

First of all, we have to identify those areas of the discretized sheet metal which lie between blank holder and die. Due to the specifications introduced in Section 2.4, all finite element nodes $\overline{p}$ with $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} \geq 65$ mm are possibly clamped; see also Figure 2.4.

The first idea was to force the z-components of all possibly clamped finite element nodes to stay unchanged. This idea is sufficient for all finite element nodes which stay within the identified areas throughout the complete simulation process. However, this finding does not hold in general for all finite element nodes since finite elements are stretched throughout the deformation process and, therefore, finite element nodes will drop out of this area. Whenever this

happens, the assumption of a constant z-component is not sufficient to rebuild the real circumstances.

A more rigorous strategy is to assume that all finite element nodes with $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} \geq 65$ mm are fixed. It is associated with blank holder forces that are so high that the material cannot move between the blank holder and the die. For sure, it is sufficient to perform the simulations exclusively on the remaining finite elements that do not belong to the predefined areas between the blank holder and the die in this case. Subsequently, we only consider finite element nodes $\overline{p}$ that fulfill $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} \leq 65$ mm and choose all finite element nodes with $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} = 65$ mm as Dirichlet boundary. This strategy works quite well for specimens with a comparable small width of the parallel shaft (less than 90 mm). However, for specimens with wider parallel shafts, it turns out that this assumption leads to an unexpected material failure in the cutoff area. This observation is also mentioned in [68], where it is described that the material fails in the cutoff area due to the prohibited material flow between the blank holder and the die. Therefore, the assumption of fully clamped sheet metals has to be slightly adapted for specimens with larger shaft widths. Due to material failing in the cutoff area, the idea is to allow material movement near the boundaries of the cutoff. As before, material movement in the remaining parts between the blank holder and the die is prohibited. More precisely, only finite element nodes $\overline{p}$ with $\sqrt{\overline{p}_x^2 + \overline{p}_y^2} \geq 65$ mm and $|\overline{p}_y| \leq 50$ mm are still assumed to be fixed; see Figure 2.4 (bottom left) and the corresponding figure in [68, Abb. 7.3] for further information. The effect of the adaptions is presented in Figure 2.5.

2.6 Exploiting Symmetry

Under ideal experimental conditions, which are considered in our numerical simulations, the macroscopic sample sheet is expected to crack perfectly along the vertical centerline. Since the experimental test setup of the Nakajima test is perfectly symmetric and DP steels are nearly isotropic, it can be advantageous to consider a half or a quarter instead of the complete sample sheet in numerical simulations.

The usage of our highly scalable software package FE2TI requires at least one individual MPI rank for each macroscopic Gauß point; see Section 4.1. Hence, the demand for individual MPI ranks for one simulation depends on the number of macroscopic integration points, which is directly connected to the number of finite elements. Since accurate finite element simulations require a sufficient discretization, the number of finite elements cannot be chosen arbi-

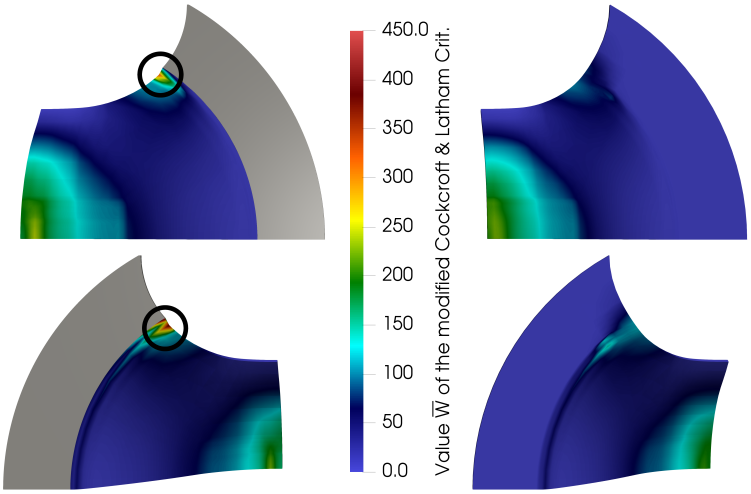


Figure 2.5: Comparison of the distribution of the value $\overline{W}$ of the modified Cockcroft & Latham criterion (see Section 2.7) for an overall tool movement of 29.303 mm with different boundary conditions. Computation of a symmetric quarter of a specimen with a width of 90 mm; see also Figure 2.4 (bottom left). **Left:** Dirichlet boundary conditions completely prohibit material flow between the blank holder and the die. Part of the specimen between the blank holder and the die is in dark grey; computed on magnitUDE. **Right:** The usage of adapted boundary conditions enables material flow between the blank holder and the die in the cutoff area. Here, we have to simulate the part of the specimen between the blank holder and the die; computed on JUWELS.

trarily. Therefore, exploiting symmetry reduces the number of individual MPI ranks without changing the discretization. The number of ranks is reduced by a factor of two when considering a half of the complete sample sheet and by a factor of four when considering a quarter of the complete sample sheet. Note that throughout this thesis, a half always means the upper half of the complete sample sheet, and a quarter means the upper right quarter; see Figure 2.6 for an example of a quarter.

By construction, only for discretizations of the complete sample sheet we can end up with a centered finite element, i.e., the center of a finite element has the same x- and y-coordinates as the center $\overline{c} = [\overline{c}_x, \overline{c}_y, \overline{c}_z]$ of the sample sheet.

Whenever we use a symmetric part of the overall sample sheet, the final solution is approximated by mirroring. Hence, to guarantee continuity of the final solution, additional partial boundary conditions have to be applied along

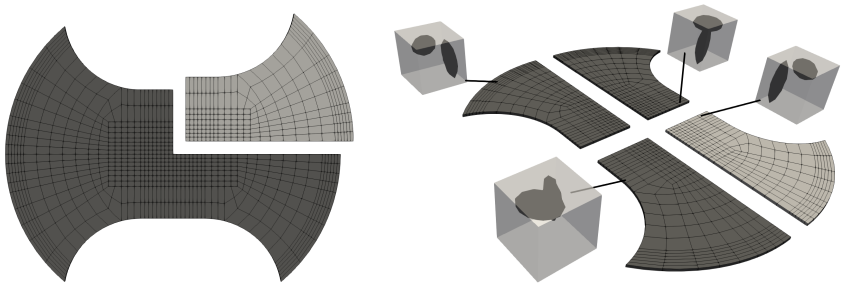


Figure 2.6: Left: Symmetric quarter (light grey) of the overall sample sheet that is used for the simulations. **Right:** Microscopic orientation change of an asymmetric SSRVE resulting from the mirroring of the symmetric solution to rebuild the overall solution. The light grey quarter is the computational domain.

the symmetric boundaries. For the half, the symmetric boundary is along the horizontal centerline. Here, the y-coordinates of all macroscopic finite element nodes $\overline{p}$ with $\overline{p}_y = 0$ have to be fixed. For a quarter, additionally displacements in x-direction have to be avoided for all macroscopic finite element nodes $\overline{p}$ with $\overline{p}_x = 0$.

Note that the use of a symmetrical part of the complete sample sheet is only expected to be exact when the RVE has a symmetric structure, since mirroring of the solution also means mirroring of the RVEs; see Figure 2.6 (right). Therefore, the assumption of a periodic unit cell is violated for symmetric computations with an asymmetric RVE. In this case, the derived solution is only an approximation to the solution using the complete sample sheet, even for the symmetric part. If we take crystal plasticity into account, which is a future goal, preferred directions along slip systems come into play (see, e.g., [129]), which cause that the macroscopic symmetry assumption generally does not hold, even for symmetric RVEs.

Except for the comparison of different RVEs (see Section 3.2), we exclusively consider an RVE with an asymmetric structure throughout this thesis; see Figure 3.8 (middle). Nonetheless, most of the simulations are carried out on a symmetric quarter because saving computing time has a higher priority than the impact on the solution.

For a comparison between simulations using a symmetric quarter and the full sample sheet we refer to Section 3.4.

2.7 Failure Criterion - a Modified Cockcroft & Latham Criterion

In the cross section method (see Section 2.1) as well as in the evaluation strategy based on thinning rates (see Section 2.2), the evaluation requires the detection of the first image for which a crack becomes visible. Since the software does not feature crack propagation, we use a phenomenological failure criterion to detect the observation of a failure zone. Therefore, a criterion similar to the Cockcroft & Latham criterion [24] is chosen. The original criterion was successfully used in a paper by Tarigopula et al. [144] for analyzing large deformation of a DP800 grade of steel. It was already introduced in 1968 and it depends on the macroscopic equivalent plastic strain $\overline{\varepsilon}^p$ and the maximum positive principal stress $\overline{\sigma}_I$ at time t_k. Note that the stress depends on the overall macroscopic strain $\bar{\epsilon}$. The original Cockcroft & Latham criterion writes

$$\widetilde{W}(\overline{\varepsilon}^p(t_k)) = \int_0^{\overline{\varepsilon}^p(t_k)} \max\left(\overline{\sigma}_I(\bar{\epsilon}(t_k)), 0\right) \; d\overline{\varepsilon}^p. \tag{2.27}$$

Since the stress tensor $\overline{\sigma}$ is a symmetric second order tensor with nine entries, it can be represented by a symmetric matrix $A \in \mathbb{R}^{3\times 3}$. The computation of the maximum principal stress component is identical to finding the maximum eigenvalue of A, which are real numbers due to its symmetry. For the computation of eigenvalues of symmetric 3×3 matrices; see Section 2.11.

Note that the deformation process is split up into several load steps in the simulations and, therefore, the evaluations at time t_k correspond to the evaluation at load step k, e.g., $\overline{\varepsilon}^p_k = \overline{\varepsilon}^p(t_k)$. Hence, Equation (2.27) can be written as

$$\widetilde{W}_k \; = \widetilde{W}(\overline{\varepsilon}^p_k) = \int_0^{\overline{\varepsilon}^p_k} \max\left(\overline{\sigma}_I(\bar{\epsilon}_k), 0\right) \; d\overline{\varepsilon}^p. \tag{2.28}$$

The failure value $\widetilde{W}_k$ in load step k is computed in each macroscopic Gauß point and is interpolated to the finite element nodes. The failure value is accumulated throughout the deformation process until a critical value $\widetilde{W}_c$ is reached, i.e., $\widetilde{W}_k \geq \widetilde{W}_c$ is fulfilled, at least in one finite element node on the top surface of the sample sheet. Exceeding the critical value is associated with the observation of a crack. In [144], a critical value between 590 MPa and 610 MPa was chosen for a DP800 steel. In the scope of this thesis, a DP600 grade of steel is considered, which requires a lower critical value $\widetilde{W}_c$ since a DP600 grade of steel is expected to be less resistant in comparison to a DP800 grade of steel.

In contrast to the original criterion due to Cockcroft & Latham [24], we use a modified version, which we denote as **modified Cockcroft & Latham criterion**. Here, we do not use the macroscopic equivalent plastic strain $\overline{\varepsilon}^p$, since the macroscopic constitutive law in the FE^2 approach is replaced by averaged microscopic values; see Section 4.1. Therefore, the average of the microscopic equivalent plastic strains $\tilde{\varepsilon}^p_k = \langle \varepsilon^p(t_k) \rangle$ is used instead of $\overline{\varepsilon}^p_k$. By usage of numerical integration, Equation (2.28) writes

$$\begin{aligned} \overline{W}_k \;=\; \widetilde{W}(\tilde{\varepsilon}^p_k) &= \int_0^{\tilde{\varepsilon}^p_k} \max\left(\overline{\sigma}_I(\overline{\epsilon}_k), 0\right) \; d\tilde{\varepsilon}^p \approx \sum_{i=1}^{k} \max\left(\overline{\sigma}_I(\overline{\epsilon}_i), 0\right) \cdot \left(\tilde{\varepsilon}^p_i - \tilde{\varepsilon}^p_{i-1}\right) \\ &= \overline{W}_{k-1} + \max\left(\overline{\sigma}_I(\overline{\epsilon}_k), 0\right) \cdot (\tilde{\varepsilon}^p_k - \tilde{\varepsilon}^p_{k-1}), \end{aligned} \tag{2.29}$$

and $\overline{W}_k$ is referred to as modified failure value in load step k. Here, $(\tilde{\varepsilon}^p_k - \tilde{\varepsilon}^p_{k-1})$ is the increment in the average of the microscopic equivalent plastic strains from load step $k-1$ to load step k and the initial values $\overline{W}_0$ and $\tilde{\varepsilon}^p_0$ are set to zero, i.e., $\overline{W}_0 = \tilde{\varepsilon}^p_0 = 0$. As in the original Cockcroft & Latham criterion, we define a critical value $\overline{W}_c$ which is associated with failure. We obtain from Equation (2.29) that the computation of the failure value $\overline{W}$ of the modified Cockcroft & Latham criterion in a specific load step requires the failure value of the last load step. Therefore, in simulations, the failure value has to be stored in each macroscopic Gauß point. Note that the failure value is updated whenever convergence of a load step is reached. For an example of the evolution of the failure criterion during the deformation process; see Figure 3.1.

2.8 Implementation of the Experimental Cross Section Method

In contrast to the real experiment, the simulation results enable the opportunity to look inside the sample sheet. Furthermore, the simulations only provide exact macroscopic values in the integration points, which in general do not coincide with the finite element nodes. Accordingly, cross sections along the top surface of the sample sheets would only take interpolated major and minor strains $\overline{\varepsilon}_1$ and $\overline{\varepsilon}_2$ into account. In contrast to the experiment, we therefore consider cross sections along Gauß points closest to the upper surface.

Due to the fully symmetric test setup of the Nakajima test, it is possible to exploit symmetry and to simulate only a symmetric part of the overall sample sheet; see Section 2.6. This automatically implies some assumptions regarding the position of the failure zone.

When we consider a quarter, the failure zone is expected to evolve along the vertical centerline and has its center on the horizontal centerline. The latter is also true if we consider a half of the full geometry, but in this case, we do not imply any assumptions regarding the horizontal position of the failure zone. Only considering the full geometry does not imply any assumptions on the position of the failure zone.

Finite element discretizations of symmetric parts of the full geometry always have finite element nodes along the horizontal centerline. Accordingly, no cross section cuts the expected vertical center of the failure zone, since we consider cross sections along integration points that usually do not coincide with finite element nodes. In addition, when considering a quarter of the full structure, we also have finite element nodes along the vertical centerline, i.e., we have no integration point at the expected horizontal center of the failure zone. Even for the simulation of the complete sample sheet, it cannot be guaranteed that we have integration points at the vertical and horizontal center of the failure zone. Therefore, we choose the first cross section along these Gauß points that have the smallest distance to the horizontal centerline. For simplicity, we also build cross sections along the remaining Gauß points of the finite elements that are used for the first cross section. Thus, the distance between the cross sections depends on the diameter of these finite elements and is generally smaller than 2 mm, which is used in the experimental cross section method. Let us note that we may have a cross section along the horizontal centerline if we do not utilize symmetry and consider the complete sample sheet. In this case, we only have two remaining rows of Gauß points in the same finite elements and we have to consider the neighboring rows of finite elements.

The choice of simulating a quarter or a half of the sample sheet or even the full geometry also affects the numerical realization of the experimental cross section method. For all symmetric computations, only cross sections above the horizontal centerline are considered, but due to the assumptions about the position of the failure zone, the cross sections below the horizontal centerline would provide identical results. Furthermore, for a quarter of the full structure, there is only one side of the cross section available, but the other side can be simply generated by mirroring; see Figure 2.2 and Figure 2.7 (top). Thus, for a quarter of the sample sheet, we have to compute the optimal fit window only for one side of the failure zone.

For the derivation of the FLD, we restrict ourselves to simulations on a quarter of the sample sheet. Unfortunately, it turns out that the failure zone evolves parallel to the vertical centerline for sample sheet geometries with a parallel

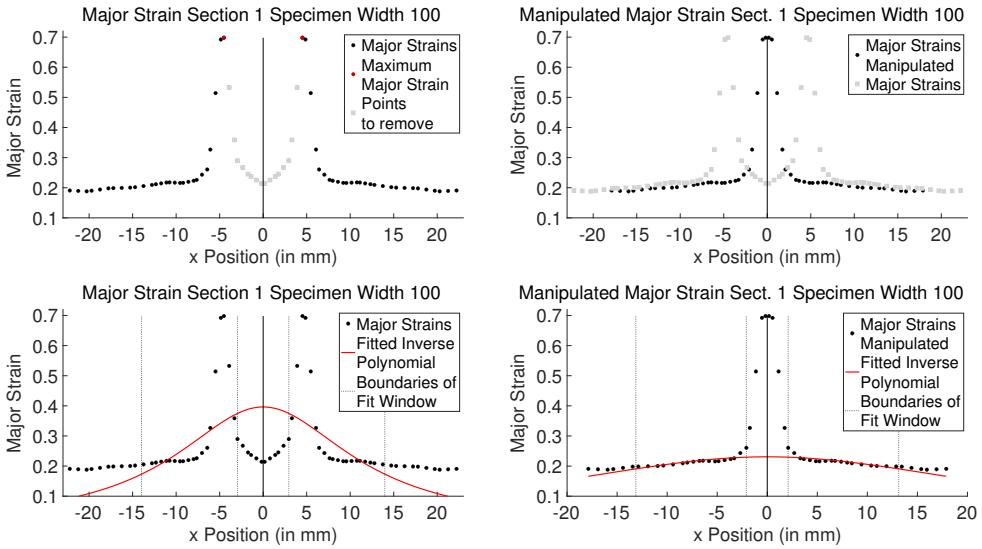


Figure 2.7: Major strains along the first cross section for the specimen with a parallel shaft width of 100 mm. **Left:** Original values of major strains and fitted inverse second-order polynomial. **Right:** Major strains along cross section after shifting maximum values to the center and fitted inverse second-order polynomial.
This figure is accepted and soon to be published; see [87, Fig. 5]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE.

shaft width of at most 100 mm. Thus, the mirroring of the solution leads to the occurrence of a second failure zone. In this case, the cross section method cannot be used as before, since it is only a valid strategy for specimens with a single crack; see [123]. Therefore, we adapt the implementation of the cross section method for simulations on a symmetric quarter with an off-centered two failure zone. We assume that the maximum major strain along the cross section defines the center of the failure zone. Neglecting all values between the vertical centerline and the maximum major strain and shifting the failure zone back to the vertical centerline, we can proceed as before; see Figure 2.7.

Of course, the final pair of major and minor strain values that is written into the FLD is strongly affected by the evaluation point and thus, it depends on the chosen critical value of the failure criterion; see Section 2.7.

2.9 Implementation of the Evaluation Based on Thinning Rates

In all our simulations, we consider a constant speed of 1 mm per second for our forming tool, i.e., each load increment makes the same (pseudo-)time increment. Following [123], the considered speed is the lower bound of allowed forming tool speeds; see also the beginning of Chapter 2.

In our simulations, we obtain a new image for each converged load step. Thus, the recording frequency in our simulations depends on the load step size. Since we have to use small load steps, especially shortly before failure, we have a higher recording frequency compared to the real test application; see Figure 2.8. Furthermore, the load increment may change during the simulation; see Section 4.2.1. Thus, the time between two consecutive images is not guaranteed to be constant, which is also in contrast to the real test application.

To overcome the problem of a higher recording frequency, the number of considered images has to be changed. Recalling that a database of 30 to 40 images is recommended, the last 3 to 4 mm of the overall tool movement are covered if a forming tool speed of $1\frac{\text{mm}}{\text{s}}$ is considered. Therefore, in our simulations, we take all images into account that belong to the last 4 mm of punch movement before reaching the critical value of the modified Cockcroft & Latham failure criterion in a finite element node on the top surface of the sample sheet; see Figures 2.3 and 2.8.

To deal with a variable pseudo time between two recorded images, the representative thinning rates are no longer plotted over the number of images but over the accumulated distance covered by the tool. To get closer to the real conditions, it is also possible to consider only images between which the forming tool has covered a distance of about 0.1 mm; see Figure 2.3 and Figure 2.8 (right).

Recalling that our simulations lead to exact values in the integration points and not in the finite element nodes, we do not compute the thinning rates at the top surface but in the Gauß points that are closest to the top surface. In contrast to the procedure introduced in [148], we have more than one Gauß point per finite element since we perform fully three-dimensional simulations. Consequently, we end up with more than one thinning rate per finite element. Throughout this thesis, we exclusively consider Q_2 finite elements on the macroscopic level and therefore, we have 9 different thinning rates per finite element. Accordingly, we first compute the average of thinning rates for each finite element before we can follow the strategy introduced in Section 2.2; see also [148].

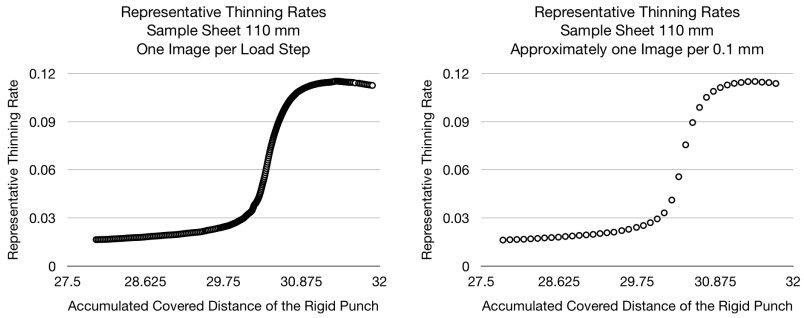


Figure 2.8: Representative thinning rates $\dot{\overline{\omega}}^k_{\text{rep}}$ for the last 4 mm before the crack appears. Sample sheet geometry with a parallel shaft width of 110 mm. **Left:** Taking into account all load steps within the last 4 mm before crack. **Right:** Consideration of load steps that have an approximative distance of about 0.1 mm.

As it is the case for the cross section method, the numerical realization of the method based on thinning rates also depends on whether a quarter, a half, or even the full sample sheet is simulated. For the full sample sheet, the variable $\alpha_{\mathcal{N}}$ should be chosen, such that five to fifteen finite elements belong to the instability zone, depending on the chosen grid size. To obtain an absolute number of about n_α finite elements in the instability zone, we have to choose $\lfloor \frac{n_\alpha}{4} \rfloor$ or $\lceil \frac{n_\alpha}{4} \rceil$ finite elements when using a quarter and $\lfloor \frac{n_\alpha}{2} \rfloor$ or $\lceil \frac{n_\alpha}{2} \rceil$ finite elements when using a half, since the full solution is approximated by mirroring.

As before (see Section 2.8), the detection of the first image that is associated with failure, i.e., the choice of the critical value $\overline{W}_c$ (see Section 2.7) strongly affects the final pair of major and minor strains. Furthermore, it may also change the characteristics of the resulting diagrams of representative thinning rates $\dot{\overline{\omega}}^k_{\text{rep}}$, $k = 1, \ldots, b$, where b is the number of images. After the linear increase with a high slope, sometimes a constant part or even a linear decrease with a small slope can be seen, especially for comparable high critical values; see Figure 2.8. Thus, the selection of the fitted linear function has to be modified. Instead of starting from the third last picture and choosing a linear function with minimal error, we generate a linear function for each n_x successive images and choose the function with maximum slope. The number n_x can be chosen by the user. The constant or linear decreasing part after the linear increase with a high slope might indicate that the chosen critical value is too high and that the specimen failed earlier.

For the computation of the velocity gradient tensor $\overline{L}$, we require the computation of the time derivative $\dot{\overline{F}}$ of the deformation gradient $\overline{F}$. It is possible

to derive $\dot{\overline{F}}$ by a linear approximation

$$\dot{\overline{F}}_t = \frac{\overline{F}_t - \overline{F}_{t-1}}{dt}, \tag{2.30}$$

where $dt = \bar{l}_t = \sum_{i=1}^{t} \bar{l}_i - \sum_{i=1}^{t-1} \bar{l}_i$ is the load increment, which is identical to the pseudo time increment. Note that $\sum_{i=1}^{t} \bar{l}_i$ is the accumulated load up to load step t. The approximative computation of $\dot{\overline{F}}$ in Equation (2.30) is assumed to be very inaccurate [148]. Therefore, a different strategy for the computation of $\dot{\overline{F}}$ is suggested in [148] and introduced in [147], where each entry of $\dot{\overline{F}}_t$ has to be computed separately using a quadratic least square fit considering seven successive load history points ranging from $t-3$ to $t+3$. During simulations, values of upcoming load steps cannot be accessed, hence the linear approximation has to be used for on line computations. Nevertheless, the least square strategy can be used when the thinning rates are computed in a post processing step.

2.10 Computation of Major and Minor Strains

The cross section method (see Section 2.1) as well as the strategy based on thinning rates (see Section 2.2) are using knowledge about the major and minor strains $\bar{\varepsilon}_1$ and $\bar{\varepsilon}_2$ on the surface of the sample sheet. Hence, the application of either the cross section method or the method based on thinning rates requires the computation of major and minor strains, at least, on the surface of the specimen.

In [148], the numerical derivation of $\bar{\varepsilon}_1$ and $\bar{\varepsilon}_2$ on the sheet metal surface is described for a finite element mesh using 4-node membrane elements. In contrast to that, we consider a fully three-dimensional simulation, i.e., we have a three-dimensional finite element discretization of the full sheet metal and not only of the surface. Therefore, we have to adapt the descriptions in [148] for our purposes.

Analogously to [148], we use a Total Lagrange description, i.e., the initial discretization of the sample sheet is used as a reference configuration and the solution of the current load step as the present configuration. The computation of $\bar{\varepsilon}_1$ and $\bar{\varepsilon}_2$ requires the two-dimensional plane strain tensor with respect to the two main directions parallel to the major and minor strain directions. Unfortunately, all quantities in our simulations are three-dimensional quantities. If we can compute the required 2D plane strain tensor $\overline{C}$ from our quantities, we can proceed as in [148].

At this point, we assume that we have successfully obtained the two-dimensional plane strain tensor $\overline{C}$; see below for the numerical implementation.

The computation of the major and minor strain values $\overline{\varepsilon}_1$ and $\overline{\varepsilon}_2$ requires the computation of the eigenvalues λ_1 and λ_2 of $\overline{C}$. Since the plane strain tensor $\overline{C}$ can be represented by a 2×2 matrix $\overline{A} = (\overline{a}_{ij})_{ij}$, $i, j = 1, 2$, the eigenvalues $\overline{\lambda}_1$ and $\overline{\lambda}_2$ can be directly computed as

$$\overline{\lambda}_1 = 0.5 \cdot \left(\overline{a}_{11} + \overline{a}_{22} + \sqrt{(\overline{a}_{11} - \overline{a}_{22})^2 + 4 \cdot \overline{a}_{12} \cdot \overline{a}_{21}} \right) \quad \text{and}$$
$$\overline{\lambda}_2 = 0.5 \cdot \left(\overline{a}_{11} + \overline{a}_{22} - \sqrt{(\overline{a}_{11} - \overline{a}_{22})^2 + 4 \cdot \overline{a}_{12} \cdot \overline{a}_{21}} \right).$$

Note, that the strain tensor is symmetric, i.e., we have $\overline{a}_{12} = \overline{a}_{21}$.

Afterwards, major and minor strains are the logarithmic strains

$$\overline{\varepsilon}_1 = \ln(1 + \overline{\lambda}_1) \quad \text{and}$$
$$\overline{\varepsilon}_2 = \ln(1 + \overline{\lambda}_2),$$

which are also called true strains or Hencky strains; see [123, 148]. The true thickness strain results from the incompressibility equation [31, Eq. 3.103]

$$\overline{\varepsilon}_3 = -(\overline{\varepsilon}_1 + \overline{\varepsilon}_2). \tag{2.31}$$

Following the arguments in [148], logarithmic strains are used due to the large strain components resulting from the Total Lagrange description.

As is suggested in [64], we use the Green-Lagrangian strain tensor $\overline{C} = 0.5 \cdot (\overline{F}^T \overline{F} - I)$, which is in contrast to [148]. As before, $\overline{F}$ is the 2D deformation gradient and I is the 2×2 identity matrix.

Derivation of the Plane Strain Tensor In the descriptions of the computation of the major and minor strain values, we have assumed that we can compute the two-dimensional plane strain tensor from three-dimensional quantities obtained from our simulations. In this section, we will briefly introduce the numerical implementation of the computation.

The starting point is the assumption that the tangent plane $\overline{T}_m$ at the mid-point $\overline{m} = [\overline{m}_x, \overline{m}_y, \overline{m}_z]$ of the upper surface of each finite element is a good approximation to its real surface. Once the corresponding tangent plane is computed, we can rotate the complete finite element such that $\overline{T}_m$ is parallel to the x-y-plane and finally project the finite element nodes of the upper surface of the rotated finite element to the tangent plane. Due to the projection, the

z-component can be neglected and the plane displacements as well as the plane strain tensor can be computed.

The tangent plane $\overline{T}_m$ is uniquely defined by its normal vector $\overline{n}^m$ and the coordinate $\overline{m}$. Note that $\overline{n}^m$ is the outward normal vector. Assuming that the surface of the sheet metal is parameterized by convective coordinates $\overline{\xi}$ and $\overline{\eta}$, the normal vector $\overline{n}^m = [\overline{n}_x^m, \overline{n}_y^m, \overline{n}_z^m]^T$ can be computed from the cross product $\overline{n}^m = \bar{t}_{\overline{\xi}} \times \bar{t}_{\overline{\eta}}$ of the tangent vectors

$$\bar{t}_{\overline{\xi}} = \left[\sum_{I=1}^{r} \frac{\partial N_I(\overline{m})}{\partial \overline{\xi}} \overline{p}_x^I, \quad \sum_{I=1}^{r} \frac{\partial N_I(\overline{m})}{\partial \overline{\xi}} \overline{p}_y^I, \quad \sum_{I=1}^{r} \frac{\partial N_I(\overline{m})}{\partial \overline{\xi}} \overline{p}_z^I\right]^T \quad \text{and}$$
$$\bar{t}_{\overline{\eta}} = \left[\sum_{I=1}^{r} \frac{\partial N_I(\overline{m})}{\partial \overline{\eta}} \overline{p}_x^I, \quad \sum_{I=1}^{r} \frac{\partial N_I(\overline{m})}{\partial \overline{\eta}} \overline{p}_y^I, \quad \sum_{I=1}^{r} \frac{\partial N_I(\overline{m})}{\partial \overline{\eta}} \overline{p}_z^I\right]^T$$

at the midpoint $\overline{m}$. Here, N_I are the basis functions belonging to the r finite element nodes $\overline{p}^I = [\overline{p}_x^I, \overline{p}_y^I, \overline{p}_z^I]$ of the upper surface of a finite element. Since we consider Q_2 finite elements, we have $r = 9$ throughout this thesis. Note that the finite element nodes $\overline{p}^I$ belong to the current state.

To determine the rotation angle $\overline{\vartheta}$, the angle between $\overline{n}^m$ and $\overline{e}_3 = [0, 0, 1]^T$ has to be computed, which yields

$$\overline{\vartheta} = \begin{cases} -\arccos(\overline{n}^{m^T} \overline{e}_3), & \overline{n}_x^m > 0, \\ -\arccos(\overline{n}^{m^T} \overline{e}_3), & \overline{n}_x^m = 0 \text{ and } \overline{n}_y^m < 0, \\ \arccos(\overline{n}^{m^T} \overline{e}_3), & \text{else.} \end{cases}$$

For the determination of the final rotation matrix $\overline{R}$, the orientation of the intersection line between the tangent plane $\overline{T}_m$ and the x-y-plane has to be determined, which is given by

$$\overline{v} = \begin{bmatrix} \overline{v}_x \\ \overline{v}_y \\ \overline{v}_z \end{bmatrix} = \begin{cases} \left[1, 0, 0\right]^T, & \text{if } \overline{n}_x^m = 0, \\ \left[-\frac{\overline{n}_y^m}{\overline{n}_x^m}, 1, 0\right]^T, & \text{otherwise.} \end{cases}$$

Finally, we can state the rotation matrix $\overline{R}$ as

$$\overline{R} = \begin{bmatrix} a\overline{v}_x^2 + \cos(\gamma) & a\overline{v}_x\overline{v}_y - \overline{v}_3 \sin(\gamma) & a\overline{v}_x\overline{v}_z + \overline{v}_y \sin(\gamma) \\ a\overline{v}_y\overline{v}_x + \overline{v}_z \sin(\gamma) & a\overline{v}_y^2 + \cos(\gamma) & a\overline{v}_y\overline{v}_z - \overline{v}_x \sin(\gamma) \\ a\overline{v}_z\overline{v}_x - \overline{v}_y \sin(\gamma) & a\overline{v}_z\overline{v}_y + \overline{v}_x \sin(\gamma) & a\overline{v}_z^2 + \cos(\gamma) \end{bmatrix},$$

with $\gamma = -\overline{\vartheta}$ and $a = 1 - \cos(\gamma)$; see [56].

To get the coordinates of the finite element nodes of the rotated finite element, we have to multiply the current coordinates with $\overline{R}$. Since only the x- and y-

components of the rotated coordinates are of interest, we can write

$$\begin{aligned} \overline{p}_x^r &= \overline{R}_{11}\overline{p}_x + \overline{R}_{12}\overline{p}_y + \overline{R}_{13}\overline{p}_z \quad \text{and} \\ \overline{p}_y^r &= \overline{R}_{21}\overline{p}_x + \overline{R}_{22}\overline{p}_y + \overline{R}_{23}\overline{p}_z \end{aligned}$$

for a specific finite element node $\overline{p} = \left[\overline{p}_x, \overline{p}_y, \overline{p}_z\right]$. The final plane displacement $\overline{u}^r$ can be derived by subtracting the reference coordinates $\overline{p}^{\text{ref}} = \left[\overline{p}_x^{\text{ref}}, \overline{p}_y^{\text{ref}}, \overline{p}_z^{\text{ref}}\right]$ of the initial state of $\overline{p}$ from the rotated coordinates $\overline{p}^r$ of the current state, i.e.,

$$\begin{aligned} \overline{u}_x^r &= \overline{p}_x^r - \overline{p}_x^{\text{ref}} \quad \text{and} \\ \overline{u}_y^r &= \overline{p}_y^r - \overline{p}_y^{\text{ref}}. \end{aligned}$$

From this displacements, the resulting plane deformation gradient $\overline{F}$ and subsequently the Green-Lagrange strain tensor $\overline{C} = 0.5 \cdot (\overline{F}^T \overline{F} - I)$ can be computed.

2.11 Computation of Eigenvalues of Symmetric 3×3 Matrices

In the scope of this thesis, we have to compute the eigenvalues of a symmetric 3×3 matrix for the evaluation of the failure criterion; see Section 2.7.

Let $A = (A_{ij}) \in \mathbb{R}^{3\times 3}$ be a symmetric matrix. Finding the eigenvalues λ of A results in finding the roots of the characteristic polynomial

$$\begin{aligned} \det(\lambda I - A) &= (\lambda - A_{11})(\lambda - A_{22})(\lambda - A_{33}) - (A_{12}A_{23}A_{31}) - (A_{13}A_{21}A_{32}) \\ &\quad - (A_{31}(\lambda - A_{22})A_{13}) - (A_{32}A_{23}(\lambda - A_{11})) - ((\lambda - A_{33})A_{21}A_{12}) \\ &= \lambda^3 - \lambda^2(A_{11} + A_{22} + A_{33}) - A_{11}A_{22}A_{33} - A_{12}A_{23}A_{31} \\ &\quad - A_{13}A_{21}A_{32} + A_{13}A_{22}A_{31} + A_{32}A_{23}A_{11} + A_{33}A_{21}A_{12} \\ &\quad + \lambda(A_{11}A_{22} + A_{11}A_{33} + A_{22}A_{33} - A_{13}A_{31} - A_{23}A_{32} - A_{12}A_{21}) \\ &= \lambda^3 - \lambda^2 \text{tr}(A) - \det(A) \\ &\quad + \lambda(A_{11}A_{22} + A_{11}A_{33} + A_{22}A_{33} - A_{13}A_{31} - A_{23}A_{32} - A_{12}A_{21}). \end{aligned} \tag{2.32}$$

Using

$$\begin{aligned} &A_{11}A_{22} + A_{11}A_{33} + A_{22}A_{33} - A_{13}A_{31} - A_{23}A_{32} - A_{12}A_{21} \\ =&\frac{1}{2}\left(2A_{11}A_{22} + 2A_{11}A_{33} + 2A_{22}A_{33} - 2A_{12}A_{21} - 2A_{13}A_{31} - 2A_{23}A_{32}\right) \\ =&\frac{1}{2}\left(\text{tr}(A)^2 - \text{tr}(A^2)\right), \end{aligned} \tag{2.33}$$

the characteristic polynomial in Equation (2.32) can be rewritten to

$$\det(\lambda I - A) = \lambda^3 - \lambda^2 \mathrm{tr}(A) - \frac{\lambda}{2}(\mathrm{tr}(A^2) - \mathrm{tr}^2(A)) - \det(A). \tag{2.34}$$

In the last equality of Equation (2.33), it was added $0 = \sum_{i=1}^{3}(A_{ii}^2 - A_{ii}^2)$.

Now, let $v \in \mathbb{R}^3$ be an eigenvector of A. If we consider the matrix $B = \frac{1}{p}(A - qI),\ p \neq 0,\ q \in \mathbb{R}$, v is also an eigenvector of B since

$$\begin{aligned} & \lambda v = Av = (pB + qI)v = pBv + qv \\ \Leftrightarrow \quad & \frac{\lambda - q}{p} v = Bv. \end{aligned}$$

Here, $\mu := \frac{\lambda - q}{p}$ is the corresponding eigenvalue of B to the eigenvector v. Hence, the eigenvalues of A can be computed from the eigenvalues of B by $\lambda = p\mu + q$. Note that B is not defined for $p = 0$, but in this case $A = pB + qI$ reduces to a diagonal matrix and therefore, the eigenvalues of A are its diagonal entries q.

Following a note in [140], we choose the parameters $q = \frac{\mathrm{tr}(A)}{3}$ and $p = \sqrt{\frac{\sum_{i=1}^{3}\sum_{j=1}^{3}(A_{ij} - qI_{ij})^2}{6}}$. Since A is symmetric, we can also write $p = \sqrt{\frac{\mathrm{tr}((A-qI)^2)}{6}}$. According to the choice of p and q, we obtain

$$\mathrm{tr}(B) = \mathrm{tr}(\frac{1}{p}(A - qI)) = \frac{1}{p}\mathrm{tr}(A) - \frac{3q}{p} = \frac{1}{p}(\mathrm{tr}(A) - \mathrm{tr}(A)) = 0, \tag{2.35}$$

$$\mathrm{tr}(B^2) = \frac{1}{p^2}\mathrm{tr}(A^2 - 2qA + q^2 I) = \frac{1}{p^2}\mathrm{tr}((A - qI)^2) = \frac{6p^2}{p^2} = 6. \tag{2.36}$$

Inserting Equations (2.35) and (2.36) into Equation (2.34) yields

$$\det(\mu I - B) = \mu^3 - 3\mu - \det(B) \tag{2.37}$$

for the characteristic polynomial of B.

Following [154, Ch. 2.2.2], Equation (2.37) has three real roots if

$$\frac{\det(B)^2}{4} - 1 \leq 0 \Leftrightarrow |\det(B)| \leq 2 \tag{2.38}$$

is fulfilled. Otherwise, Equation (2.37) has one real root and two imaginary roots.

By definition, B is real and symmetric. Hence, the roots of Equation (2.37) are real numbers and, therefore, Equation (2.38) is fulfilled, since

$$0 \text{ im. roots} \Rightarrow\ !\ (2 \text{ im. roots}) \Rightarrow\ !\ (|\det(B)| > 2) \Rightarrow |\det(B)| \leq 2$$

holds. Defining $\tilde{b} := \frac{1}{3}\arccos\left(\frac{\det(B)}{2}\right)$, the eigenvalues of B are given by

$$\begin{aligned}\mu_1 &= 2\cos(\tilde{b}),\\ \mu_2 &= -\cos(\tilde{b}) - \sqrt{3}\sin(\tilde{b}), \quad \text{and}\\ \mu_3 &= -\cos(\tilde{b}) + \sqrt{3}\sin(\tilde{b});\end{aligned}$$

see [154, Ch. 2.2.2] and the corresponding errata [153]. We can reformulate μ_2 and μ_3 to

$$\mu_2 = 2\cos(\tilde{b} + \frac{2\pi}{3}) \qquad \text{and} \qquad \mu_3 = 2\cos(\tilde{b} + \frac{4\pi}{3})$$

by using the equality

$$\cos\left(\frac{\arccos\left(\frac{\det(B)}{2}\right)}{3} + \frac{2k\pi}{3}\right) =$$
$$\cos\left(\frac{\arccos\left(\frac{\det(B)}{2}\right)}{3}\right)\cos\left(\frac{2k\pi}{3}\right) - \sin\left(\frac{\arccos\left(\frac{\det(B)}{2}\right)}{3}\right)\sin\left(\frac{2k\pi}{3}\right),\ k = 1, 2, ;$$

see [154, Ch. 6.5.9].

Finally, the eigenvalues of A write $\lambda_i = q + p \cdot \mu_i,\ i = 1, 2, 3.$

3 Numerical Results for the Simulation of the Nakajima Test

In this chapter, we have collected some results regarding the simulations of the Nakajima test with different sample sheets that were performed on the supercomputers

- JUWELS at Jülich Supercomputing Centre, Germany; European Tier 0; TOP500 rank 23 in the year 2018 (114 480 cores; 9.8 petaflops); main source of compute time for the computation of an FLD; see Section 3.1
- MagnitUDE (Tier-3): 13 536 cores (Broadwell XEON E5-2650v4 12C 2.2GHz; 24 cores and 72 GB per node); 476.5 TFlops NEC Cluster; operated by Center for Computational Sciences and Simulation (CCSS) of the Universität Duisburg-Essen (UDE) providing computing resources for UDE; TOP500 rank 384 (November, 2016).

In order to simulate the deformation process of a DP steel accurately, we have to take into account an elasto-plastic material model; see Section 2.3 and the references therein. Furthermore, also the choice of the representative volume element (RVE) has an impact on the deformation behavior; see Section 3.2. For the derivation of an RVE representing a realistic microstructure, electron backscatter diffraction is used; see [17]. For DP steels, the martensitic inclusions in the ferrite matrix are quite small and widely spread. Thus, a realistic microstructure of DP steels requires a fine discretization of the RVE, which leads to large microscopic problems. To overcome this problem, we do not take into account an RVE with a realistic microstructure, but we consider statistically similar RVEs (SSRVEs); see [8, 133]. Instead of considering the small martensitic islands, the SSRVEs only consider inclusions of simple geometries such as ellipsoids, but describe the overall mechanical behavior of the DP steel appropriately. The final shapes of the inclusions are derived by an optimization process. As a result, the inclusions in the SSRVEs are much simpler compared to the realistic microstructure and, therefore, we can use a coarser grid for the discretization. If not noted otherwise, we always consider an SSRVE with two ellipsoidal inclusions that was fitted to the realistic microstructure of a DP600

grade of steel; see Figure 3.8 (middle). It is discretized using 1 470 P_2 finite elements in an unstructured manner resulting in 7 152 degrees of freedom.

Throughout this thesis, we exclusively use structured Q_2 finite elements for the discretization of the macroscopic sample sheet geometries and, if not noted otherwise, we use two finite elements in thickness direction. The number of finite elements depends on the width of the parallel shaft as well as on whether we use a symmetric part of the sample sheet or not; see Sections 2.5 and 2.6. Let us note that both, the macroscopic as well as the microscopic discretizations are derived by using the open source software package GMSH [49].

We provide an individual MPI rank for each microscopic boundary value problem, i.e., for each macroscopic integration point. Since we consider hexahedral Q_2 finite elements, we have 27 integration points per macroscopic finite element. Consequently, the number of required MPI ranks can be directly computed by multiplying the number of finite elements with 27.

In the simulation of the Nakajima test, we consider a contact-driven deformation that results from the movement of the rigid punch in upward direction; see Figure 2.1. As one can see in Table 3.1, the rigid punch has to move about 30 mm until failure is detected. Of course, this load is much too large to apply it in one step. Consequently, we split the movement of the rigid punch into smaller load steps. As an initial load step, we always choose $\bar{l}_{\text{init}} = 0.1$ mm. In our simulations, we make use of a dynamic load step strategy (see Section 4.2.1) and a linear extrapolation strategy (see Section 4.2.2). Accordingly, the load step size is expected to change various times during the simulation process. To exclude too large load steps, we always prescribe a maximum allowed load step size $\bar{l}_{\text{max}} = 0.2$ mm. The linear extrapolation is activated after the third load step.

Since friction between the rigid punch and the deformable body has to be avoided as much as possible in the Nakajima test (see [123]), we consider frictionless contact based on a penalty formulation; see Section 4.2.4. If not stated otherwise, we have chosen a penalty parameter $\bar{\varepsilon}_N = 500$ in our simulations.

The derivation of the virtual FLD requires the simulation of the Nakajima test for different sample sheets. Following [123], at least five different sample sheets are required for a valid FLD. Nonetheless, we have used 10 different sample sheets, including a fully circular specimen and sample sheets with parallel shaft widths of 30 mm, 50 mm, 70 mm, 90 mm, 100 mm, 110 mm, 125 mm, and 129 mm; see Section 2.4 and Figure 2.4. For the derivation of the virtual FLD, we have restricted ourselves to the usage of the symmetric quarter. On one hand, the computational costs are reduced by a factor of about 4. On the other

hand, the resulting problem sizes can be solved on mid-sized HPC systems, which seems reasonable for the application in industries.

In our software, we have not implemented crack propagation. Instead, we use a phenomenological failure criterion, namely the modified Cockcroft & Latham criterion, for the detection of a crack; see Section 2.7. Unfortunately, we do not have experimental data to calibrate a critical value $\overline{W}_c$, which is associated with the appearance of a crack. The original Cockcroft & Latham criterion was used in [144] for a DP800 grade of steel and they have provided a critical value of $\overline{W}_c = 590 - 610$ MPa. Based on this, we have estimated a critical value $\overline{W}_c = 450$ MPa, since a DP600 grade of steel is less robust compared to a DP800 grade of steel. However, to ensure that we do not stop the simulation too early, the stopping criterion in our simulations is not based on this critical value. Accordingly, the simulation proceeds if the critical value is reached in a macroscopic finite element node on the top surface of the sample sheet, even if it is associated with failure.

Since the stopping criterion is not based on the failure criterion, we have to formulate other conditions to terminate the simulation. On one hand, the simulation ends if the overall load, i.e., the accumulated rigid tool movement, reaches the corresponding value that was prescribed by the user. On the other hand, the stopping criterion is also based on the load step size. If the load step size of 10 consecutive load steps is smaller than a predefined allowed minimum load step size, the simulation somehow stagnates and is therefore stopped. Furthermore, if the load step size has to be reduced seven times within a single load step, the simulation also terminates. A stopping criterion based on the load step size is motivated by the fact that small load steps indicate hard numerical problems, which are surely expected in the case of failure. In all our computations, the minimum allowed load step size is $10^{-4} \cdot \bar{l}_{\text{init}} = 10^{-5}$ mm.

If not noted otherwise, we always use the sparse direct solver package MKL PARDISO [132] for solving the resulting tangent problems on both scales. As interval between two checkpoints, we have always used 75 load steps.

3.1 A Virtual Forming Limit Diagram

First, we present the results of different production runs for the derivation of the virtual FLD and the corresponding FLCs based on the different evaluation strategies introduced in Sections 2.1 and 2.2; see also Sections 2.8 and 2.9. For the computation of the virtual FLD, we have exclusively performed our simulations on the JUWELS supercomputer [70]. Some details on the simulations

Table 3.1: Some details on Nakajima simulations with different sample sheets. Microscopic problems: SSRVE with two ellipsoidal inclusions; 1 470 unstructured P2 finite elements and 7 152 d.o.f; see Figure 3.8 (middle). Two MPI ranks per core; computed on JUWELS [70]; one microscopic problem computed per MPI rank. Overall problem size is obtained by multiplying the number of MPI ranks by d.o.f. of microscopic SSRVE and adding the number of macroscopic d.o.f. resulting in 80 - 112 million d.o.f.
This table is in more detail accepted and soon to be published; see [87, Tab. 2]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE..

	width 30	**width 50**	**width 70**	**width 90**	**width 100**	**width 110**	**width 129**	**Full Circular**
Macro Finite Elements (Q_2)	424	460	482	558	558	580	574	542
MPI Ranks	11 448	12 420	13 014	15 066	15 066	15 660	15 498	14 634
Macro d.o.f.	13 725	14 804	15 465	17 835	17 835	18 495	18 195	17 145
Covered Dist. Punch (mm)	27.156	29.242	29.896	30.734	31.654	32.593	36.566	40.000
Load Steps	736	806	901	985	898	780	1 651	569
Newton Its.	8 148	9 272	9 823	10 976	8 604	8 540	10 012	8 064
Runtime (h)	17.60	19.35	22.00	27.00	24.00	21.00	24.17	19.25
Restarts	1	1	1	2	1	1	1	1
Overhead Load Steps	74	6	8	68	16	24	3	12
Overhead Runtime (h)	1.70	0.23	0.23	1.71	0.40	0.58	0.25	0.41

are provided in Table 3.1, including the time to solution, the number of macroscopic finite elements for the discretization of the sample sheet, and the final movement of the rigid punch. In addition, we also present the number of required restarts, which are mostly caused by reaching the pre-chosen wall time limit. For all our simulations, we have prescribed a maximum movement of the rigid punch of 40 mm.

We obtain similar simulation results for all sample sheets with a parallel shaft width. If the rigid punch moves beyond a certain distance, we observe a strong local increase of different parameters, such as the failure values $\overline{W}$ of the modified Cockcroft & Latham criterion, the major strains $\bar{\varepsilon}_1$, the equivalent plastic strains $\tilde{\varepsilon}^p$, the von Mises stresses, and the thinning rates $\dot{\bar{\omega}}$. Due to the local increase in the failure values $\overline{W}$ of the modified Cockcroft & Latham criterion, the local area is associated with the failure zone. Within this range, the thickness of the sample sheet drastically reduces; see Figure 3.2 and Figure 3.3 (left) as well as [87, Fig. 10]. For wider parallel shaft widths, the rigid tool has to move

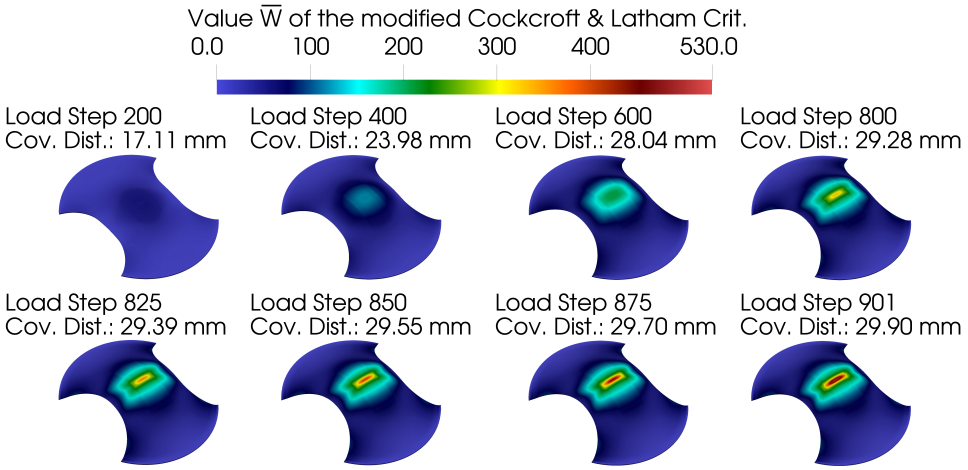


Figure 3.1: Evolution of the failure values $\overline{W}$ of the modified Cockcroft & Latham criterion (see Section 2.7) on the top surface of a sample sheet with a parallel shaft width of 70 mm; symmetric quarter; see also [87, Fig. 9] for the evolution of $\overline{W}$ for a sample sheet with a parallel shaft width of 50 mm. Computed on JUWELS [70]. Fur further information regarding the simulation; see the corresponding column in Table 3.1.

further to force the local increase; see Table 3.1. Obviously, all simulations considering sample sheets with a parallel shaft terminate before the prescribed maximum distance is reached. For a better understanding how the different parameters evolve during the simulation process, we present the evolution of the modified failure values $\overline{W}$ during the simulation process for a sample sheet with a parallel shaft width of 70 mm; see Figure 3.1. We observe, that the modified failure values $\overline{W}$ drastically increase in a small area within the last 100 load steps, even if they make just a small increment in the accumulated covered distance of the forming tool.

Although we observe the evolution of localized failure zones for all sample sheets with a parallel shaft, the position of the failure zone varies. For all sample sheets with a parallel shaft width of at least 90 mm, the failure zone evolves along the vertical centerline, as it is expected since we consider a symmetric quarter. However, for all sample sheets with wider parallel shafts, we obtain off-centered failure zones; see Figure 3.3 (left). Recent results indicate that the off-centered failure zones are caused by the symmetry assumption; see Section 3.4 for further details. Nevertheless, we use the results for the generation of FLDs. For the evaluation strategy based on thinning rates, the position of the failure

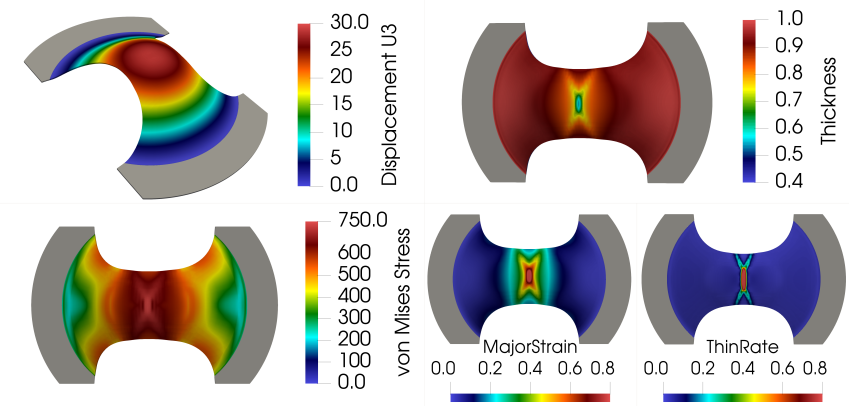


Figure 3.2: Final simulation result of the Nakajima test using a sample sheet with a parallel shaft width of 50 mm; displacements in z-direction (top left), thickness (top right), von Mises stresses (bottom left), major strains (bottom center), and thinning rates (bottom right). Dirichlet boundary conditions prohibit material movement between the blank holder and the die (dark grey). This part is not simulated. Computed on JUWELS [70].

zone is irrelevant, and for the cross section method, we slightly manipulate the data as mentioned in Section 2.8.

For the fully circular sample sheet, we do not see strong localized effects, even if it is the only sample sheet for which the rigid punch moves 40 mm in upward direction. For this reason, we cannot apply the method based on thinning rates for this specimen, since the thinning rates only slightly increase; see Figure 3.4. Of course, it is quite similar for the major strains along the cross sections; see Figure 2.2 (right). However, the cross section method yields evaluation points, but their physical meanings are questionable. Nonetheless, we reach the critical value $\overline{W}_c = 450$ MPa in finite element nodes on the top surface of the sample sheet, which is associated with failure; see Figure 3.3 (right).

The final FLDs obtained from the cross section method and the method based on thinning rates as well as the corresponding FLCs are presented in Figures 3.5 and 3.6, respectively. For the computation of the FLCs as well as for a comparison of the results obtained by using the different methods, we refer to Section 3.1.1.

3.1.1 Computation of Forming Limit Curves

The main characteristics of a forming limit diagram are similar for all different types of steel. The wider the parallel shaft width of the sample sheet, the larger

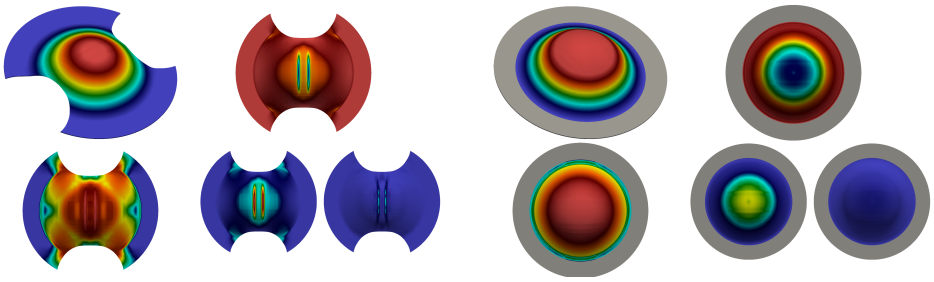


Figure 3.3: Final simulation results of the Nakajima test using a sample sheet with a parallel shaft width of 100 mm (**left**) and the completely circular specimen (**right**); computed on JUWELS [70]; variables and color bars as in Figure 3.2. **Left:** Material between blank holder and die is simulated since material movement is allowed. **Right:** Dirichlet boundary conditions prohibit material movement between the blank holder and the die (dark grey). This part is not simulated.
This figure is accepted and soon to be published; see [87, Fig. 11]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE.

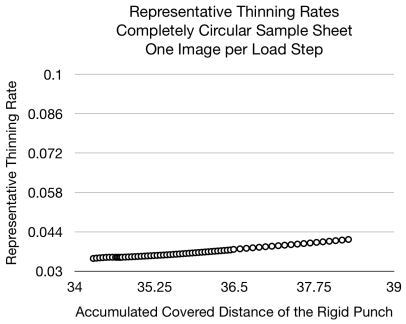


Figure 3.4: Representative thinning rates $\dot{\overline{\omega}}^k_{\text{rep}}$ considering the completely circular sample sheet and the last 4 mm before the critical value $\overline{W}_c$ is exceeded.

are the values of the minor strains. The sample sheet, which corresponds to the minimum major strain, divides the pairs of major and minor strains in the FLD into two parts. On one hand, we obtain a nearly linear decrease in the major strain values for all sample sheets which have a smaller parallel shaft width than the sample sheet which belongs to the minimum major strain, i.e., for all pairs of major and minor strains which are placed to the left of the minimum

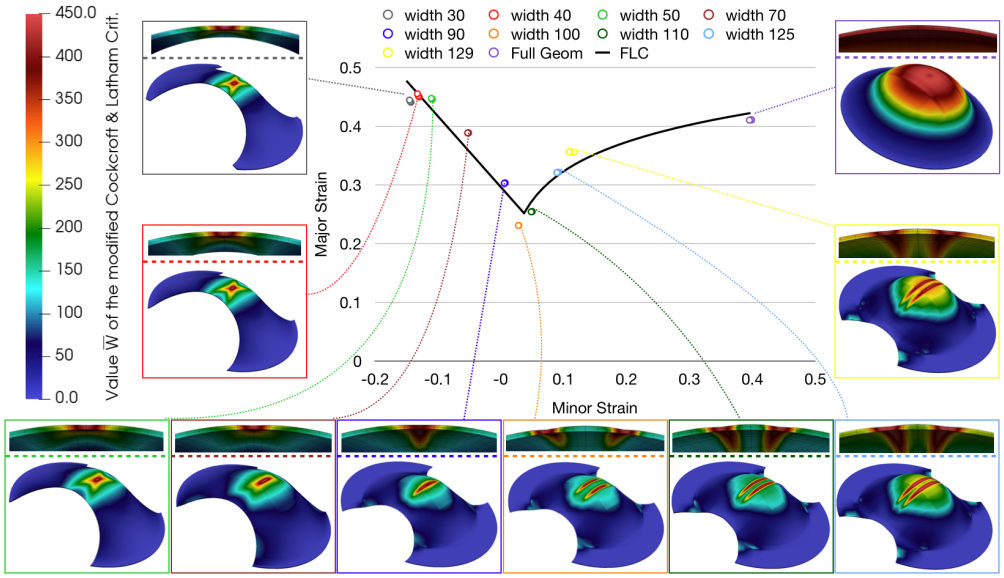


Figure 3.5: FLD with FLC (black curve) for $\overline{W}_c = 450$ MPa obtained from the cross section method. Distribution of failure values $\overline{W}$ of the modified Cockcroft & Latham criterion for all Nakajima simulations immediately after reaching the critical value $\overline{W}_c$ on the top surface. In the cross sections, we identify local necking in thickness for all but the full circular specimen; computed on JUWELS [70]. This figure is accepted and soon to be published; see [87, Fig. 12]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE.

major strain. On the other hand, the major strain values show a logarithmic increase for all sample sheets which have a wider parallel shaft compared to the sample sheet which belongs to the minimum major strain, i.e., for all pairs which are placed to the right of the minimum major strain.

Due to the characteristics of an FLD, we derive the forming limit curve by the combination of two regression functions $f_l(x)$ and $f_r(x)$, each belonging to one characteristic part of the FLD. The left part $f_l(x)$ is obtained from a simple linear regression, i.e., $f_l(x) = m \cdot x + n$. The right part $f_r(x)$ is obtained by a least squares fit to a logarithmic function, i.e., $f_r(x) = a \cdot \ln(x) + b$. Finally, we have to determine the intersection point x_{int} of both parts, which is

$$x_{\text{int}} = -\frac{a \cdot W\left(-\frac{m \cdot e^{b/a+n/a}}{a}\right)}{m},$$

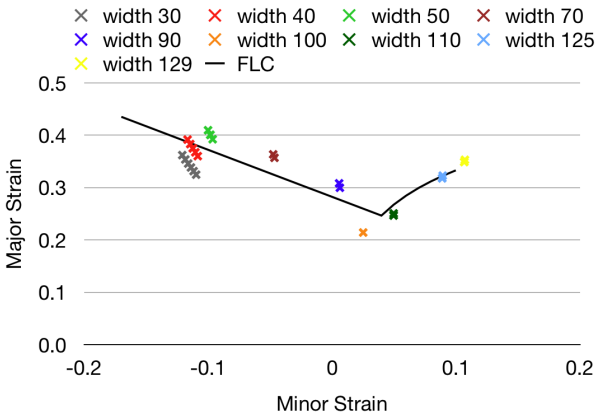


Figure 3.6: FLD and corresponding FLC (black curve) for $\overline{W}_c = 450$ MPa obtained from the evaluation based on thinning rates. We have used the same colors as in Figure 3.5 for the different sample sheet geometries.

where $W(x)$ is the product logarithm or Lambert W function [25].

Finally, the FLC writes

$$\text{FLC} = \begin{cases} f_l(x), & x \in [x_{\min}, x_{\text{int}}], \\ f_r(x), & x \in (x_{\text{int}}, x_{\max}], \end{cases}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum minor strains of all points in the FLD. The final FLC is written into the FLD; see the black curves in Figures 3.5 and 3.6.

Note that the points of the sample sheet belonging to the minimum major strain value are used in the regression of both parts. In our case, the sample sheet with a parallel shaft width of 100 mm belongs to the minimum major strain; see Figures 3.5 and 3.6.

Taking into account the different evaluation strategies, i.e., the cross section method (see Sections 2.1 and 2.8) and the evaluation based on thinning rates (see Sections 2.2 and 2.9), we obtain different FLDs and therefore different FLCs. The FLD as well as the final FLC obtained from the cross section method are shown in Figure 3.5 and were already presented in [87]. The recent implementation of the evaluation method based on thinning rates allows us to present a second FLD with its corresponding FLC; see Figure 3.6.

For remarks regarding the implementation of both evaluation strategies, we refer to Sections 2.8 and 2.9, respectively.

For the evaluation based on thinning rates, we only consider load steps with a tool movement of approximately 0.1 mm between them. Consequently, for the determination of the critical finite elements, we especially do not consider the second last load step before the critical value was reached in a finite element node on the top surface of the sample sheet, but the load step for which the tool was driven up at least 0.2 mm less than at the time of failure. The variable $\alpha_{\mathcal{N}}$ is chosen such that 8 or 12 finite elements of the overall sample sheet belong to the critical area; see Section 2.2. Since we consider a symmetric quarter, we have to detect 2 or 3 finite elements with highest average thinning rates.

For the computation of the last image before the occurrence of plastic instability, we have to fit interpolation polynomials to both parts that are characterized by a linear increase; see, e.g., Figure 2.8. For the approximation of the unstable area, we compute a linear polynomial using a least squares fit with five consecutive images each. At the end, we choose that polynomial with the maximum slope. Since the final slope may depend of the number of images that are considered within the least squares fit, we also consider other slopes. Therefore, we compute the intersection angle of the computed polynomial with the x-axis. Afterwards, we increase and decrease the intersection angle by 5 %, 10 %, 15 %, and 20 %, which leads to 8 other polynomials. Accordingly, we derive 9 different evaluation points for the FLD.

It turns out that both evaluation strategies yield quite similar evaluation points for sample sheets with a parallel shaft width of at least 70 mm. Only for the smallest parallel shaft widths of 30 mm and 40 mm, the evaluation based on thinning rates leads to smaller major strain and larger minor strain values; see Figure 3.7 (left). In addition, rotating the approximation polynomials to the unstable area has a greater effect to the final values than for the remaining sample sheet geometries; see Figure 3.6 and Figure 3.7 (left).

For a better comparison of the resulting FLCs, we have neglected the full sample geometry, since the evaluation strategy based on thinning rates cannot be applied for this case. Due to the nearly identical evaluation points obtained for sample sheet geometries with a parallel shaft width of at least 100 mm, it is not surprising that the logarithmic interpolations are nearly the same for both evaluation strategies. The left part of the FLD obtained from the cross section method is approximated by a linear polynomial. Therefore, we also approximate the left part of the FLD obtained from the method based on thinning rates with a linear polynomial, even if the arrangement of evaluation points looks different; see Figure 3.7. For the smallest sample sheet geometries, the evaluation points obtained from the method based on thinning rates are below the evaluation

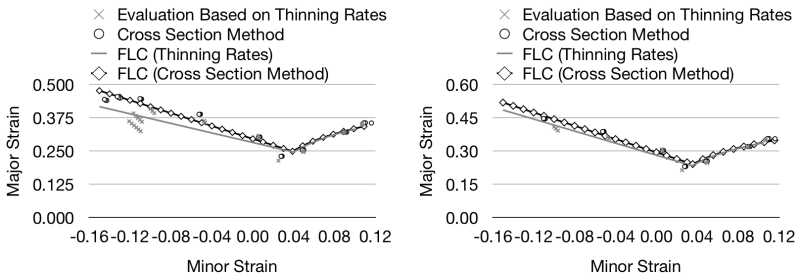


Figure 3.7: Comparison of the resulting FLCs obtained from the cross section method and the evaluation strategy based on thinning rates using different subsets of the simulation results; see Sections 2.1, 2.2, 2.8 and 2.9. **Left:** All sample sheets with a parallel shaft. **Right:** Sample sheets with a parallel shaft width of at least 50 mm.

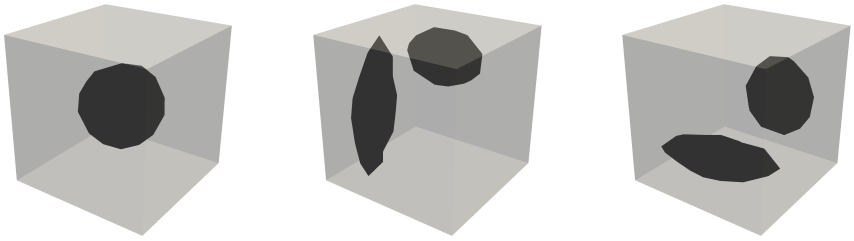


Figure 3.8: Different types of SSRVEs. The SSRVE in the middle is used for the computation of the FLD and is called "Standard SSRVE" throughout this thesis.

points obtained from the cross section method. Accordingly, the resulting linear approximation polynomial leads to larger values for the cross section method; see Figure 3.7 (left). However, if we neglect the evaluation points belonging to the sample sheet geometries with a parallel shaft width of 30 mm and 40 mm, the resulting linear approximations are again similar; see Figure 3.7 (right).

3.2 Impact of Different SSRVEs

In this section, we study the impact of the considered SSRVE on the simulation behavior as well as on the final evaluation point. Therefore, we compare simulations considering the same computational domain but different SSRVEs. Apart from this section, we have exclusively used the microstructure that is shown in the middle of Figure 3.8, which is referred to as standard SSRVE in this section. Accordingly, we compare one of the two other SSRVEs with the standard SSRVE. We do not only look at the final simulation results, but

Table 3.2: Comparison of the usage of different SSRVEs for a sample sheet geometry with a parallel shaft width of 40 mm; see Figure 2.4 (top left). Here, we consider the standard SSRVE (computed on JUWELS [70]) and an SSRVE with a spherical inclusion (computed on magnitUDE); see Figure 3.8. The resulting major and minor strain values $\bar{\varepsilon}_1^{\mathrm{FLC}}$ and $\bar{\varepsilon}_2^{\mathrm{FLC}}$ are obtained from the evaluation using the cross section method; see Sections 2.1 and 2.8. Note that the evaluation is independent of the simulation results beyond reaching the critical value $\overline{W}_c = 450$ MPa for finite element nodes on the top surface of the sample sheet.

Comparison of standard SSRVE and a spherical inclusion Sample sheet with a parallel shaft width of 40 mm					
		Standard SSRVE		Spherical Inclusion	
		$\overline{W}_c \geq 450$ (top surface)	Final step	$\overline{W}_c \geq 450$ (top surface)	Final step
Load Step		710	751	762	805
Max. $\overline{W}$ (top surface) [MPa]		451.54	563.02	450.56	502.97
Cov. Dist. Punch [mm]		27.813	28.325	28.194	28.412
∅ Load Step Size [mm]		0.039	0.038	0.037	0.035
∅ Load Step Size beyond $\overline{W} \geq \overline{W}_c$ [mm]		0.012		0.005	
Major Strain $\bar{\varepsilon}_1^{\mathrm{FLC}}$	Sec. 1	0.450		0.455	
	Sec. 2	0.451		0.456	
	Sec. 3	0.455		0.458	
Minor Strain $\bar{\varepsilon}_2^{\mathrm{FLC}}$	Sec. 1	-0.130		-0.127	
	Sec. 2	-0.131		-0.128	
	Sec. 3	-0.133		-0.128	

also on the corresponding results immediately after reaching the critical value $\overline{W}_c = 450$ MPa for the first time for a finite element node on the top surface.

At first, let us consider a very simple SSRVE with a spherical inclusion (see Figure 3.8 (left)). Therefore, we deal with a sample sheet with a shaft width of 40 mm; see Figure 2.4 (top left). In Table 3.2, we present some relevant information of both simulations, including the overall covered distance of the forming tool and the resulting evaluation point obtained from the cross section method. Furthermore, we have also plotted the differences of the displacements in z-direction as well as of the modified failure values $\overline{W}$ and the major strains $\bar{\varepsilon}_1$ on the top surface of the sample sheet immediately after reaching the critical value in at least one finite element node on the top surface of the sample sheet; see Figure 3.9. No matter which SSRVE is used, the accumulated covered distance of the forming tool is quite similar at the

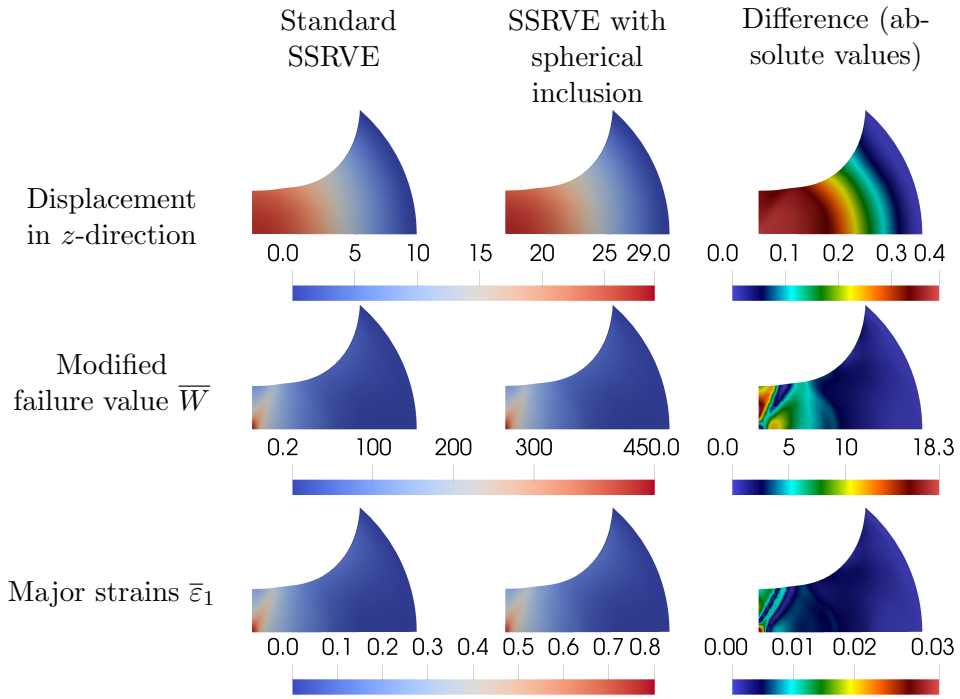


Figure 3.9: Comparison of the z-displacements (top), the failure values $\overline{W}$ of the modified Cockcroft & Latham criterion (middle), and the major strains $\overline{\varepsilon}_1$ (bottom) on the top surface of a sample sheet with a parallel shaft width of 40 mm immediately after reaching the critical value $\overline{W}_c = 450$ MPa in a finite element node on the top surface of the sample sheet. We consider the standard SSRVE (left) and a simple spherical inclusion (middle). For further information; see Table 3.2.

end; see Table 3.2. However, the resulting maximum values of the modified Cockcroft & Latham criterion differ by about 10 %. Even if the accumulated covered distance of the rigid punch is a bit larger for the SSRVE with a single spherical inclusion, the maximum final value of the modified Cockcroft & Latham criterion on the top surface of the sample sheet is smaller. In addition, to reach the critical value $\overline{W}_c$ at the top surface of the sample sheet, the rigid tool also has to move further in comparison to the usage of the standard SSRVE; see Table 3.2. Consequently, using the SSRVE with a single spherical inclusion leads to a material that can be deformed to a larger extend without failing. Since the FLC characterizes the transition between permitted and prohibited deformations, it is not surprising that the resulting major strain value, which is written into the FLD, is a bit larger; see Table 3.2. Due to the very small differences in the major strain values next to the failure zone

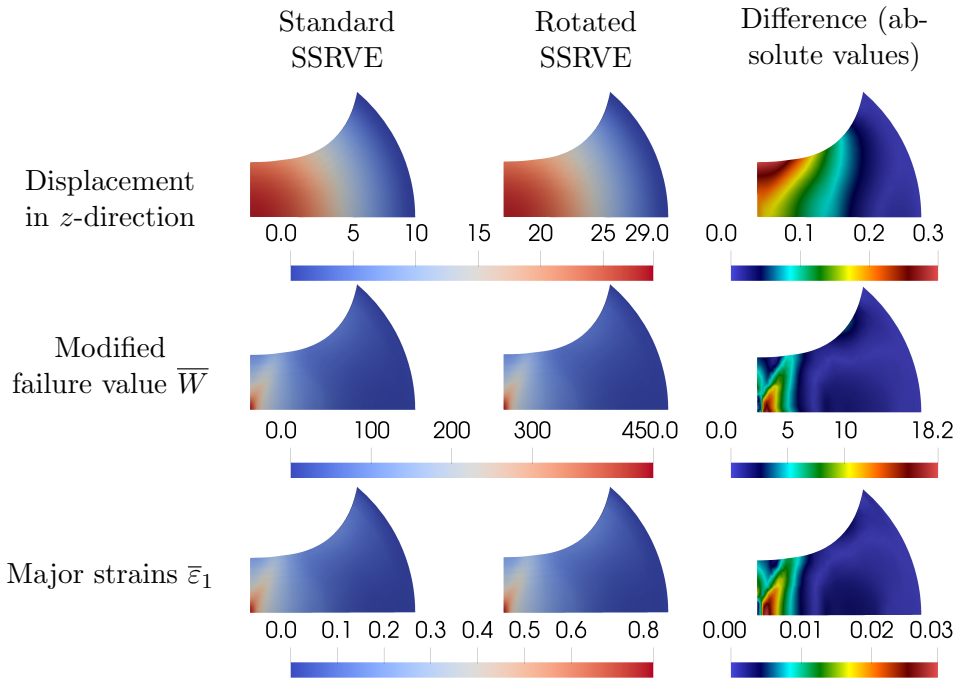


Figure 3.10: Comparison of the z-displacements (top), the failure values $\overline{W}$ of the modified Cockcroft & Latham criterion (middle), and the major strains $\bar{\varepsilon}_1$ (bottom) on the top surface of a sample sheet with a parallel shaft width of 50 mm immediately after reaching the critical value $\overline{W}_c = 450$ MPa in a finite element node on the top surface of the sample sheet. We consider the standard SSRVE (left) and a rotated version of it (middle). For further information; see Table 3.3.

(see Figure 3.9), the final major strain values obtained from the cross section method only slightly differ; see Table 3.2.

For a comparison of the differently rotated SSRVEs with identical ellipsoidal inclusions, we consider a sample sheet geometry with a parallel shaft width of 50 mm; see Figure 2.4 (top right). Again, we provide the same information as before; see Table 3.3 and Figure 3.10. We obtain that the rotated SSRVE reaches the critical value $\overline{W}_c$ earlier, i.e., after a smaller accumulated covered distance of the rigid punch; see Table 3.3. With the same arguments as in the previous paragraph, the corresponding point in the FLC moves downwards if we consider the rotated SSRVE instead of the standard SSRVE. Note that the resulting difference is more significant compared to the difference between the standard SSRVE and the spherical inclusion due to the larger differences in the major strain values next to the failure zone; see Figure 3.10.

Table 3.3: Comparison of the usage of different SSRVEs for a sample sheet geometry with a parallel shaft width of 50 mm; see Figure 2.4 (top right). Here, we consider the standard SSRVE and a rotated version of it; see Figure 3.8. Both simulations have been computed on JUWELS [70]. For further remarks see the caption of Table 3.2.

Comparison of the standard SSRVE and a rotated version Sample sheet with a parallel shaft width of 50 mm					
		Standard SSRVE		Rotated SSRVE	
		$\overline{W}_c \geq 450$ (top surface)	Final step	$\overline{W}_c \geq 450$ (top surface)	Final step
Load Step		743	806	742	781
Max. $\overline{W}$ (top surface) [MPa]		451.33	571.63	452.69	489.40
Cov. Dist. Punch [mm]		28.747	29.242	28.566	28.713
∅ Load Step Size [mm]		0.039	0.036	0.038	0.037
∅ Load Step Size beyond $\overline{W} \geq \overline{W}_c$ [mm]		0.008		0.004	
Major Strain $\overline{\varepsilon}_1^{\text{FLC}}$	Sec. 1	0.446		0.416	
	Sec. 2	0.446		0.416	
	Sec. 3	0.447		0.418	
Minor Strain $\overline{\varepsilon}_2^{\text{FLC}}$	Sec. 1	-0.109		-0.097	
	Sec. 2	-0.110		-0.097	
	Sec. 3	-0.110		-0.096	

Even if the deformation behavior seems to be similar until the critical value is reached, the usage of the rotated SSRVE seems to be numerically harder beyond this stage. After reaching the critical value, we perform less load steps until the simulation terminates, even if the average load step size is only half as large; see Table 3.3. As a result, using the standard RVE seems to be less affected by the fact that the simulation goes beyond the critical value $\overline{W}_c$ that is associated with failure. Accordingly, the final simulation results differ to a larger extent; see Figure 3.10.

Surprisingly, also the SSRVE with a spherical inclusion seems to be numerically harder than the standard SSRVE after reaching the critical value; see Table 3.2.

3.3 Some Tests on the Penalty Parameter

In this section, we show the effects of different penalty parameters $\overline{\varepsilon}_N$ on the solutions as well as on the required computing times. For our test, we consider a sample sheet geometry with a parallel shaft width of 40 mm. For all our simulations, we move the rigid punch 10 mm in upward direction, while we take into

Table 3.4: Computational information using different penalty parameters for a sample sheet geometry with a parallel shaft width of 40 mm (see Figure 2.4 (top left)); computed on magnitUDE.

	Sample sheet 40 computed on magnitUDE; 2 MPI ranks per core			
	$\overline{\varepsilon}_N = 50$	$\overline{\varepsilon}_N = 500$	$\overline{\varepsilon}_N = 5\,000$	$\overline{\varepsilon}_N = 50\,000$
Cov. Dist. Punch [mm]	10	10	10	10
Load Steps	74	101	129	198
Newton Its.	656	1 015	1 236	1 854
∅ Load Step Size [mm]	0.135	0.099	0.078	0.051
∅ Newton Its. per Load Step	8.86	10.05	9.58	9.36
Runtime [s]	5 347.60	8 090.79	9 847.62	14 511.90
∅ Time per Load Step [s]	72.26	80.11	76.34	73.29
∅ Time per Newton It. [s]	8.15	7.97	7.97	7.83
# penetrated FE nodes	220	180	141	109
Max. Pen. [mm]	0.22	0.04	0.01	0.002

account different penalty parameters. Some details on the different simulations are presented in Table 3.4, including the time to solution, the average load step size, and the maximum penetration of a finite element node resulting from the rigid punch. For all finite element nodes that have a larger distance d to the center of the rigid tool than the radius $r_{\mathcal{T}}$ of the rigid tool, the penetration is set to zero. For the remaining finite element nodes, the penetration computes as $r_{\mathcal{T}} - d$; see also Section 4.2.4.

It turns out that smaller penalty parameters lead to significantly faster runtimes, which are affected by the larger average load step sizes. However, the maximum amount of penetration is much larger. In addition, also the number of penetrated finite element nodes is larger; see also Figure 3.11 for the illustration of all finite element nodes that penetrate into the rigid punch. These observations are consistent with the literature; see, e.g., [151]. We note that the faster computing times for smaller penalty parameters probably result from the smaller total deformations, which allow larger load steps.

Throughout this thesis, we have used a penalty parameter of 500. This can be motivated by a smaller runtime without significantly increasing the maximum amount of penetration compared to larger penalty parameters.

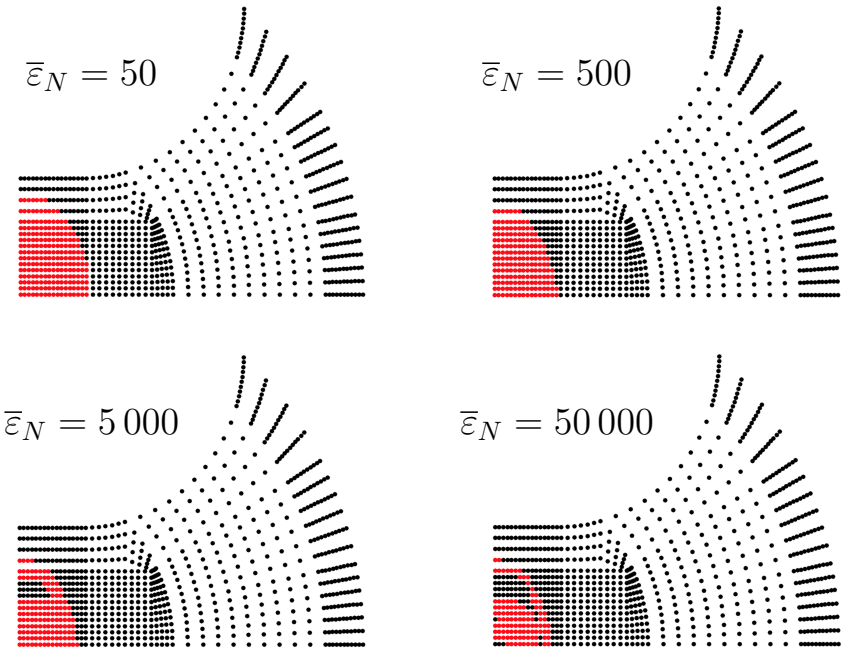


Figure 3.11: Comparison of penetrated FE nodes for different penalty parameters $\overline{\varepsilon}_N$ for a covered distance of the rigid punch of 10 mm; symmetric quarter; sample sheet geometry with a parallel shaft width of 40 mm; computed on magnitUDE. For further information; see Table 3.4.

In the future, we plan to introduce an option to start with a comparably small penalty parameter that can be increased during the simulation. It will be interesting to see, how the computing times and solutions compare to simulations with a constant penalty parameter.

3.4 Influence of Symmetry to the Final Solution

In this section, we compare simulation results obtained from using a symmetric quarter with the results obtained from simulations considering the complete sample sheet. Since the failure zone evolves perfectly symmetric for all sample sheets with a parallel shaft width of at most 90 mm, we focus on the geometries with off-centered failure zones using a symmetric computation. To be more specific, we exclusively consider a sample sheet with a parallel shaft width of 100 mm throughout this section.

For our comparison, we have used different discretizations for the symmetric quarter as well as for the complete sample sheet. For the symmetric quarter, we deal with two discretizations, which differ only in the number of finite ele-

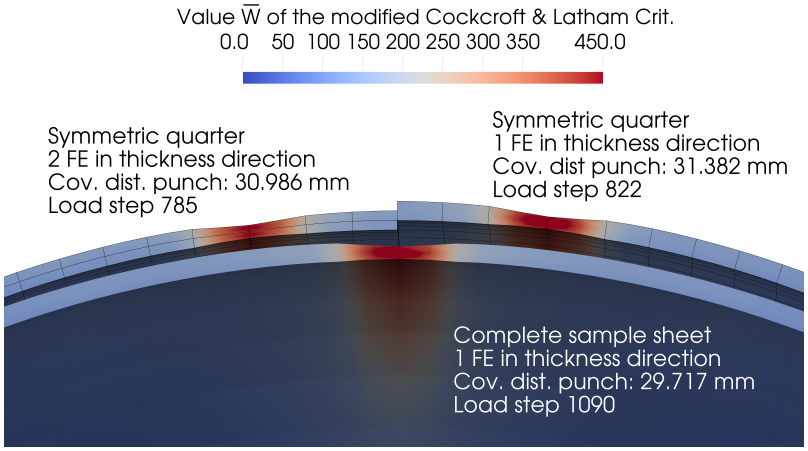


Figure 3.12: Comparison of the failure values $\overline{W}$ of the modified Cockcroft & Latham criterion for the simulation results immediately after reaching the critical value $\overline{W}_c = 450$ MPa in finite element nodes on the top surface of the sample sheet. We consider symmetric computations with one and two finite elements in thickness direction as well as the corresponding discretization of the complete sample sheet with one finite element in thickness direction for a sample sheet with a parallel shaft width of 100 mm. Different heights of the sample sheets results from different tool movements. Even if we present the upper left quarter for the symmetric computation with two finite elements in thickness direction, we have computed the same symmetric quarter as mentioned in Section 2.6. All three simulations were performed on JUWELS [70] and we have used the NK-BDDC approach for the parallel solution of the macroscopic problem for the non-symmetric simulation.

ments in thickness direction. Here, we consider one and two finite elements in thickness direction. Besides the corresponding discretizations of the complete sample sheet, we also take into account a third discretization with one finite element in thickness direction and larger finite elements compared to the other discretizations.

For both simulations using the symmetric quarter, we get comparable results. Immediately after reaching the critical value $\overline{W}_c = 450$ in at least one finite element node on the top surface of the sample sheet, the overall movement of the rigid punch is slightly higher for the finite element discretization with one finite element in thickness direction, but we obtain the same off-centered failure zone for both simulations ; see Figure 3.12.

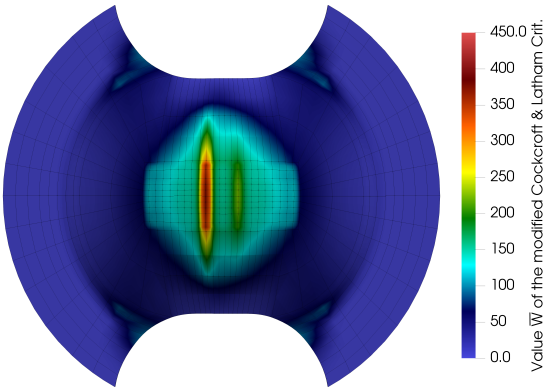


Figure 3.13: Final distribution of the modified failure value $\overline{W}$ for a simulation without utilizing symmetry for a sample sheet with a parallel shaft width of 100 mm; computed on magnitUDE. Note that the chosen discretization is coarser compared to the corresponding discretization of the symmetric quarter that was used for the simulation of the FLD; see also Figure 3.12.

.

For the coarsest discretization of the complete sample sheet, we also get an off-centered failure zone, which fits to the results using a symmetric quarter; see Figure 3.13. However, taking into account a finer discretization (in x- and y-direction), which is equivalent to the discretization of the symmetric quarter with one finite element in thickness direction, the failure zone evolves along the vertical centerline; see Figure 3.12. In comparison to the symmetric computation, the critical value $\overline{W}_c$ is exceeded for a smaller movement of the rigid punch; see the different heights in Figure 3.12.

We obtain similar results for the discretization of the complete sample sheet with two finite elements in thickness direction. Unfortunately, the simulation has stopped before the critical value $\overline{W}_c = 450$ MPa was reached due to small load steps that have not been allowed so far. However, we observe a local formation of a failure zone along the vertical centerline for the last load step. For a better comparison, we also provide similar simulation states of the corresponding symmetric computation; see Figure 3.14. The application of the identical load using a symmetric quarter does not lead to strong localized effects; see Figure 3.14 (top right). Therefore, we also consider the simulation state corresponding to a similar modified failure value $\overline{W}$, which corresponds to a rigid tool movement of about 0.5 mm further; see Figure 3.14 (bottom right). In fact, at this time, we observe a localized evolution of an off-centered failure zone; see again Figure 3.14 (bottom right).

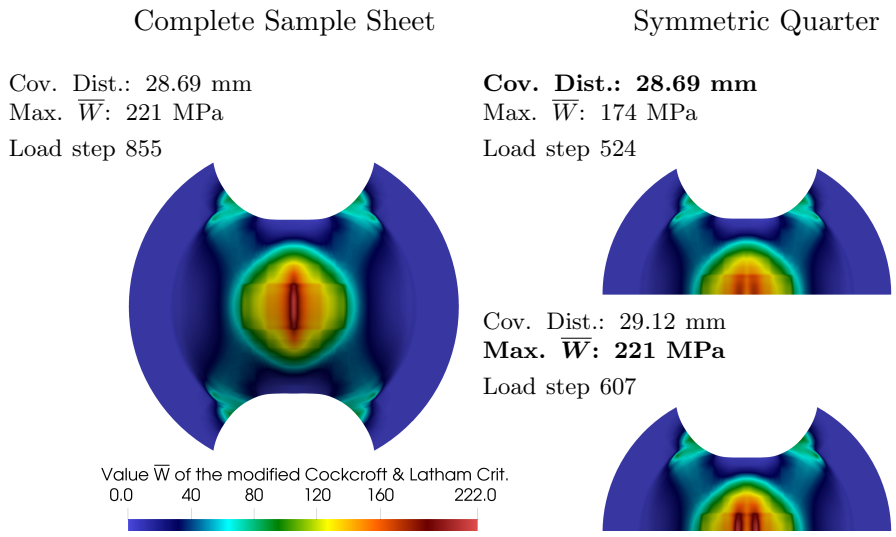


Figure 3.14: Comparison of the values $\overline{W}$ of the modified Cockcroft & Latham criterion for symmetric and non-symmetric simulations for a sample sheet with a parallel shaft width of 100 mm and a discretization with two finite elements in thickness direction. Both simulations have been performed on JUWELS [70] using equivalent discretizations. For the non-symmetric simulation, we have used the NK-BDDC approach for the macroscopic problem.

It turns out that the symmetric computations seem to have an impact on the position of the evolution of the failure zone. Therefore, we have to take into account the complete sample sheet instead of a symmetric quarter, at least for all sample sheets with off-centered failure zones. Moreover, the simulation of the fully circular sample sheet might also lead to a localized failure zone without utilizing symmetry. As we have presented for the sample sheet with a parallel shaft width of 100 mm, we have to enable the usage of smaller load steps for simulations on the complete sample sheets. In addition, we can also think about further options for the dynamic load step strategy in order to prevent critical load steps.

3.5 Newton-Krylov BDDC for the Macroscopic Problem

We have recently incorporated the Newton-Krylov BDDC approach (see Section 5.6.1) for the parallel solution of the macroscopic problems in our FE2TI software. As a result, we are able to efficiently solve larger macroscopic problems. For some simulation results using the NK-BDDC approach on the macroscopic level, we refer to Section 3.4, where we have used the BDDC approach

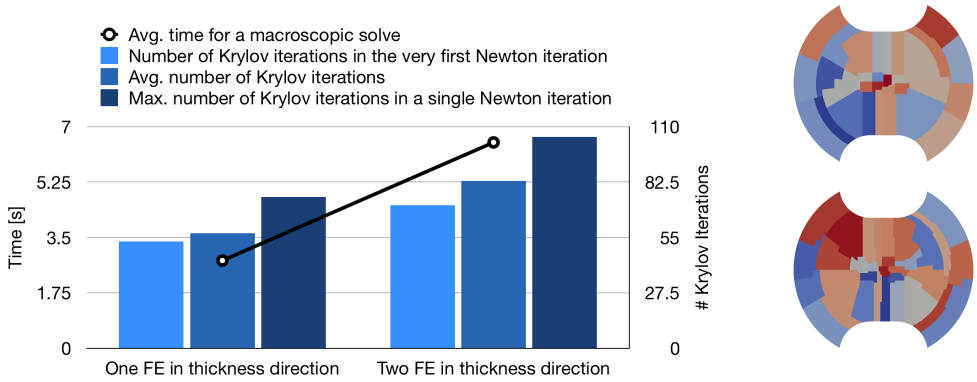


Figure 3.15: Left: Iteration numbers of the NK-BDDC approach for solving the macroscopic problem in a Nakajima simulation considering different discretizations of the complete sample sheet with a parallel shaft width of 100 mm. For further information, such as the resulting shape of the sample sheet, we refer to Section 3.4; computed on JUWELS [70]. **Right:** Decomposition of the discretizations of the complete sample sheet with a parallel shaft width of 100 mm into 32 subdomains. The discretizations with one (top) and two (bottom) finite elements in thickness direction are considered in Sections 3.4 and 3.5.

for the simulations of the complete sample sheet. The focus of this section is rather the numerical analysis of FE2TI using NK-BDDC than the quality of the computed solutions.

As it is standard for domain decomposition methods, good scalability is usually only achieved if a suitable coarse problem is incorporated. In our simulations, we have used the following coarse space for the BDDC method. Besides the subdomain vertices, the subassembly is also performed in a single finite element node of each edge on the subdomain interface, because the number of subdomain vertices is comparably small due to the sample sheet geometry. Furthermore, for each face across the interface, we have chosen additional constraints following the frugal approach in [58]. Due to the macroscopic homogeneity, this is equivalent to weighted averages along faces. For both simulations on the complete sample sheet that are considered in Section 3.4, the coarse problem contains about 400 degrees of freedom.

It turns out that the NK-BDDC approach performs quite well for our problems. As a first indicator, we look at the number of Krylov iterations which are needed for the solution of the macroscopic problem. Whenever we use the

NK-BDDC approach as a solver for the macroscopic problem in FE2TI, we exclusively use GMRES as a Krylov subspace method. Both, the flat sample sheet geometry and the corresponding METIS decomposition are expected to have a negative effect on the performance of a Krylov subspace method. Nonetheless, the number of Krylov iterations is in an acceptable range from the very beginning; see Figure 3.15 (left). However, this can be improved by using a better coarse space, e.g., by choosing an adaptive coarse space; see, e.g., [75, 91, 106, 117, 141, 142]. Moreover, the number of Krylov iterations per macroscopic Newton iteration only slightly increases during the simulation. As a result, the NK-BDDC approach is quite robust and requires more or less constant time for the solution, no matter how many finite elements belong to the plastic regime.

The increased average time for the solution of the macroscopic problem in a single Newton iteration (see Figure 3.15 (left)) can be explained by the fact that we have decomposed the considered discretizations into the same number of subdomains; see Figure 3.15 (right) for the decomposition of both discretizations. Since we have used discretizations with one and two finite elements in thickness direction, the overall number of finite elements is twice as large for the latter case. Consequently, also the subdomains are approximately twice as large. As a result, the average time is expected to increase with a factor of more than 2, since we use a sparse direct solver for the subdomain problems. For our simulations, we obtain a factor of about 2.35, which is satisfactory.

4 FE^2: Theory and Implementation

For the simulation of the Nakajima test, we use our highly scalable software package FE2TI, which combines a parallel implementation of the FE^2 approach [43,103,121,134,135,138] and different domain decomposition methods such as FETI-DP [38,39,93,94,97,98] and BDDC [26,32,111,115,116] as solver on both levels, the macroscale and the microscale. It is a C/C++ implementation based on PETSc [4–6] and MPI.

We first give a short introduction to the FE^2 approach. Afterwards, we introduce the software package FE2TI and give some remarks on recent software developments such as the implementation of a contact formulation (see Section 4.2.4) on the macroscopic level and a dynamic load step strategy (see Section 4.2.1). This chapter is based on the discussion in [87]

4.1 The FE^2 Method

The macroscopic material properties of many materials, such as DP steels, result from their heterogeneous microscopic structure. Therefore, accurate finite element simulations require the incorporation of these microscopic heterogeneities.

A brute force finite element discretization of the macroscopic problem down to the microscopic level is not feasible due to two different reasons. On one hand, the resulting system of equations would be too large to solve it, even on the largest available supercomputers. On the other hand, the microstructure of the overall macroscopic domain is usually not known.

Instead, the incorporation of the microstructure has to be achieved with other techniques. In our simulations, we consider the FE^2 method that is a computational homogenization approach. In the following, we introduce the necessary equations for the FE^2 method as well as the numerical implementation. It is mainly based on the presentations in [135], but also on [16,107].

In the FE^2 approach, the macroscopic and the microscopic scale are discretized separately. The macroscopic problem is discretized completely ignoring the microscopic structure, i.e., the problem is assumed to be homogeneous from a macroscopic point of view. The characteristic length scale of the macroscopic problem is denoted by L. We assume that the microscopic heterogeneities can

be represented by a volume fraction of the overall microstructure, which is called representative volume element (RVE). Its length scale is denoted by l in the following. For the incorporation of the microscopic heterogeneities, we solve an independent microscopic boundary value problem in each macroscopic integration point. Let us note that the same RVE is used for all macroscopic Gauß points, but the boundary values of the microscopic problems depend on the macroscopic deformation gradient in the corresponding integration point. On the macroscopic level, a phenomenological material law is replaced by volumetric averages of microscopic stresses. In addition, we also have to compute a consistent tangent modulus that takes into account the volumetric average of the tangent moduli of the corresponding microscopic problem; see, e.g. [135]. Accordingly, constitutive models for the different phases have to be set up exclusively on the microscopic scale. The applicability of the FE^2 approach requires a scale separation which is assumed to be fulfilled considering DP steels, i.e., $L \gg l$ holds. For an illustration of the basic ideas of the FE^2 approach we refer to [74, Fig. 1].

As before, macroscopic quantities will be marked with an overline to distinguish them from microscopic quantities. For example, the macroscopic deformation gradient is denoted by $\overline{F}$.

We denote the reference configuration of the macroscopic domain by $\overline{\mathcal{B}}_0 \subset \mathbb{R}^3$ and the current configuration by $\overline{\mathcal{B}} \subset \mathbb{R}^3$. Material points in the reference configuration are denoted by $\overline{X}$ and material points in the current configuration are denoted by $\overline{x}$. The deformation $\overline{\varphi} : \overline{\mathcal{B}}_0 \to \overline{\mathcal{B}}$ maps points from the reference configuration to the current configuration and the macroscopic deformation gradient $\overline{F}(\overline{X})$ is defined by

$$\overline{F}(\overline{X}) := \mathrm{Grad}_{\overline{X}} \left(\overline{\varphi}(\overline{X}) \right) .$$

With the macroscopic first Piola-Kirchhoff stress tensor $\overline{P}$ and some external load $\overline{f}$, the linear balance of momentum with respect to the reference configuration writes

$$\mathrm{Div}_{\overline{X}} \overline{P} + \overline{f} = 0.$$

In the same manner as for the macroscopic problem, we can formulate microscopic quantities, i.e., the microscopic deformation $\varphi : \mathcal{B}_0 \to \mathcal{B}$ maps material points X from the reference configuration of an RVE to material points x in the actual configuration. Analogously to the macroscopic deformation gradient,

the microscopic deformation gradient writes

$$F(X) := \mathrm{Grad}_X\left(\varphi(X)\right).$$

The linear balance of momentum with respect to the reference configuration for the microscopic scale writes

$$\mathrm{Div}_X P = 0.$$

Here, we neglect external forces.

The macroscopic first Piola-Kirchhoff stresses $\overline{P}$ are computed as volumetric average of the microscopic first Piola-Kirchhoff stresses P, i.e.,

$$\overline{P} := \frac{1}{V}\int_{\mathcal{B}_0} P(F)\ dV, \tag{4.1}$$

where $V = |\mathcal{B}_0|$ is the volume of the reference configuration of the RVE. The same holds for the macroscopic deformation gradient

$$\overline{F} := \frac{1}{V}\int_{\mathcal{B}_0} F\ dV.$$

The latter two equations can also be written as surface integrals; see [135, Eq. (34)] and [16, Eq. (7.6)].

The macro-homogeneity condition, also known as Hill's condition or Hill-Mandel condition, is one of the most important relations in micro-macro scale bridging schemes and writes

$$\overline{P} : \dot{\overline{F}} = \frac{1}{V}\int_{\mathcal{B}_0} P : \dot{F}\ dV. \tag{4.2}$$

Here, $\dot{\overline{F}}$ and $\dot{F}$ denote the time derivatives of the macroscopic as well as of the microscopic deformation gradient, respectively, and can be written as

$$\dot{\overline{F}} = \mathrm{Grad}_{\overline{X}}\dot{\overline{x}}, \qquad\qquad \dot{F} = \mathrm{Grad}_X \dot{x}. \tag{4.3}$$

Following [135], Hill's condition can be reformulated to

$$\begin{aligned}
& \frac{1}{V}\int_{\mathcal{B}_0} (P-\overline{P}) : (\dot{F}-\dot{\overline{F}})\ \mathrm{d}V = 0 \\
\Leftrightarrow \quad & \frac{1}{V}\int_{\mathcal{B}_0} (P-\overline{P}) : \dot{F}\ \mathrm{d}V - \frac{1}{V}\int_{\mathcal{B}_0} (P-\overline{P}) : \dot{\overline{F}}\ \mathrm{d}V = 0 \\
\Leftrightarrow \quad & \frac{1}{V}\int_{\mathcal{B}_0} (P-\overline{P}) : \mathrm{Grad}_X \dot{x}\ \mathrm{d}V - \frac{1}{V}\int_{\mathcal{B}_0} (P-\overline{P}) : \dot{\overline{F}}\mathrm{Grad}_X X\ \mathrm{d}V = 0,
\end{aligned} \tag{4.4}$$

where the last equation is obtained by inserting Equation $(4.3)_2$ and the identity $I = \mathrm{Grad}_X X$. Note that the second term on the left-hand side can also be written as

$$\frac{1}{V}\int_{\mathcal{B}_0} \dot{\overline{F}}^T (P - \overline{P}) : \mathrm{Grad}_X X \; \mathrm{d}V.$$

Now, we reformulate both parts on the left-hand side of Equation (4.4) using $\dot{x}^T P = (P^T \dot{x})^T$ and $\mathrm{Div}(P\dot{x}) = P^T : \mathrm{Grad}\dot{x} + \dot{x} \cdot \mathrm{Div} P^T$ (see [29, Sec. 2.5.8, Eq. (v)]). For the first part, we obtain

$$\int_{\mathcal{B}_0} (P - \overline{P}) : \mathrm{Grad}_X \dot{x} \; \mathrm{d}V = \int_{\mathcal{B}_0} \mathrm{Div}_X(\dot{x}^T (P - \overline{P})) - \dot{x}^T \mathrm{Div}_X (P - \overline{P}) \; \mathrm{d}V.$$

The second term on the right-hand side vanishes due to the equilibrium requirement $\mathrm{Div}_X(P - \overline{P}) = 0$; see [135]. Applying the Gaussian integral theorem (see [47, Sec. 15]) to $\int_{\mathcal{B}_0} \mathrm{Div}_X(\dot{x}(P - \overline{P})) \; \mathrm{d}V$, we obtain

$$\begin{aligned}
\int_{\mathcal{B}_0} (P - \overline{P}) : \mathrm{Grad}_X \dot{x} \; \mathrm{d}V &= \int_{\partial\mathcal{B}_0} (\dot{x}^T (P - \overline{P})) \cdot N \; \mathrm{d}A \\
&= \int_{\partial\mathcal{B}_0} \dot{x}^T \left((P - \overline{P}) \cdot N \right) \; \mathrm{d}A \\
&= \int_{\partial\mathcal{B}_0} \left((P - \overline{P}) N \right)^T \dot{x} \; \mathrm{d}A, \qquad (4.5)
\end{aligned}$$

where N is the outward normal vector to $\partial\mathcal{B}_0$. Analogously, the second term on the left-hand side of Equation (4.4) writes

$$\int_{\mathcal{B}_0} (P - \overline{P}) : \dot{\overline{F}} \mathrm{Grad}_X X \; \mathrm{d}V = \int_{\partial\mathcal{B}_0} \left((P - \overline{P}) N \right)^T \dot{\overline{F}} X \; \mathrm{d}A. \qquad (4.6)$$

Here, the term $\mathrm{Div}_X(\dot{\overline{F}}(P - \overline{P}))$ vanishes since $\dot{\overline{F}}$ is independent of X and because of the equilibrium requirement $\mathrm{Div}_X(P - \overline{P}) = 0$. Finally, combining Equations (4.5) and (4.6) and using Cauchy's theorem (see, e.g., [29, Sec. 3.3.1] and [14, Ch. 6, Th. 1.3]), i.e., inserting $t = PN$, we obtain

$$\frac{1}{V}\int_{\partial\mathcal{B}_0} (t - \overline{P}N)^T (\dot{x} - \dot{\overline{F}} X) \; dA = 0, \qquad (4.7)$$

which is equivalent to Equation (4.2); see also [135].

From [135, Ch. 2.5], we obtain that Equation (4.7) also holds if the time derivatives $\dot{x}$ and $\dot{\overline{F}}$ are replaced by the original values x and $\overline{F}$ or δx and δF, respectively. With this in mind, different boundary conditions can be directly read from Equation (4.7). We obtain valid Dirichlet-type boundary conditions

by choosing $x = \overline{F}X \ \forall X \in \partial\mathcal{B}_0$, since it fulfills Equation (4.7) when $\dot{x}$ and $\dot{\overline{F}}$ are replaced by x and $\overline{F}$. Alternatively, the choice $t = \overline{P}N \ \forall X \in \partial\mathcal{B}_0$ fulfills Equation (4.7), which is associated with a Neumman-type boundary condition.

In addition, a third type of suitable boundary conditions can be derived - the periodic boundary conditions. Therefore, the overall boundary $\partial\mathcal{B}_0$ has to be subdivided into two different parts $\partial\mathcal{B}_0^+$ and $\partial\mathcal{B}_0^-$, where each point $X^+ \in \partial\mathcal{B}_0^+$ has an associated partner $X^- \in \partial\mathcal{B}_0^-$. We introduce a periodic fluctuation field $\tilde{w}$ which fulfills $x = \overline{F}X + \tilde{w}$. Following [135], a periodic fluctuation field is defined by $\tilde{w}^+ = \tilde{w}^- \ \forall X^+ \in \partial\mathcal{B}_0^+$, $X^- \in \partial\mathcal{B}_0^-$, and the corresponding outward normals N^+ and N^- fulfill the relation $N^+ = -N^-$. Replacing $\dot{x}$ and $\dot{\overline{F}}$ in Equation (4.7) by x and $\overline{F}$ and subsequently using the definition of the periodic fluctuation field $\tilde{w} = x - \overline{F}X$, we obtain

$$
\begin{aligned}
& \frac{1}{V}\int_{\partial\mathcal{B}_0} (t - PN)\cdot\tilde{w}\ dA = 0 \\
\Leftrightarrow\quad & \frac{1}{V}\left(\int_{\partial\mathcal{B}_0^+} (t^+ - PN^+)\cdot\tilde{w}^+\ dA + \int_{\partial\mathcal{B}_0^-} (t^- - PN^-)\cdot\tilde{w}^-\ dA\right) = 0 \\
\Leftrightarrow\quad & \frac{1}{V}\int_{\partial\mathcal{B}_0^+} (t^+ - PN^+ + t^- + PN^+)\cdot\tilde{w}^+\ dA = 0 \\
\Leftrightarrow\quad & \frac{1}{V}\int_{\partial\mathcal{B}_0^+} (t^+ + t^-)\cdot\tilde{w}^+\ dA = 0.
\end{aligned}
$$

From the last equation, we obtain the necessary condition $t^+ = -t^-$ for a periodic fluctuation field $\tilde{w}$. Let us note that [16, Ch. 7.1.2] also refers to an alternative derivation in [121].

The numerical implementation requires the weak formulation of the balance of momentum on both scales. On the microscopic scale, the weak form writes

$$
-\int_{\partial\mathcal{B}_0} \delta x \cdot \mathrm{Div}_X(P(F))\ dV = 0, \tag{4.8}
$$

where δx is a variational function. Analogously, the weak form of the macroscopic balance of momentum writes

$$
-\int_{\partial\overline{\mathcal{B}}_0} \delta\overline{x} \cdot \left(\mathrm{Div}_{\overline{X}}(\overline{P}(\overline{F})) + \overline{f}\right)\ dV = 0,
$$

where $\delta\overline{x}$ is a variational function.

In general, the resulting systems of equations on both scales have a nonlinear character. Therefore, the solutions are derived by an iterative process. Here, we make use of Newton's method. Thus, we have to compute the linearization of

the resulting systems. On both scales, the computation of the stiffness matrices of a single finite element requires the computation of the tangent modulus in each integration point, which is the partial derivative of the stresses with respect to the deformation gradient. As mentioned before, we only have a constitutive material law on the microscopic level. Thus, the macroscopic tangent moduli $\overline{\mathbb{A}} = \frac{\partial \overline{P}}{\partial \overline{F}}$ cannot be computed directly.

In the following, we focus on periodic boundary conditions on the microscopic level. Recalling the representation $x = \overline{F}X + \tilde{w}$, where $\tilde{w}$ is a periodic fluctuation field, we obtain the additive decomposition of the microscopic deformation gradient $F = \overline{F} + \widetilde{F}$. Note that the macroscopic deformation gradient $\overline{F}$ is known and constant over the RVE attached to the corresponding macroscopic integration point.

Recalling from Equation (4.1) the relation of the macroscopic and microscopic stresses $\overline{P} = \frac{1}{V}\int_{\mathcal{B}_0} P(F)\ \mathrm{d}V$, the macroscopic tangent modulus writes

$$\overline{\mathbb{A}} = \frac{\partial \overline{P}}{\partial \overline{F}} = \frac{\partial}{\partial \overline{F}}\left(\frac{1}{V}\int_{\overline{\mathcal{B}}_0} P(F)\ \mathrm{d}V\right) = \frac{1}{V}\int_{\overline{\mathcal{B}}_0} \frac{\partial P(F)}{\partial \overline{F}}\ \mathrm{d}V.$$

Inserting $F = \overline{F} + \widetilde{F}$ into the latter equation, we obtain with the chain rule

$$\begin{aligned}
\overline{\mathbb{A}} &= \frac{1}{V}\int_{\partial\overline{\mathcal{B}}_0} \frac{\partial P(F)}{\partial \overline{F}}\ dV \\
&= \frac{1}{V}\int_{\partial\overline{\mathcal{B}}_0} \frac{\partial P(F)}{\partial F} : \frac{\partial F}{\partial \overline{F}}\ dV \\
&= \frac{1}{V}\int_{\partial\overline{\mathcal{B}}_0} \mathbb{A} : \frac{\partial(\overline{F} + \widetilde{F})}{\partial \overline{F}}\ dV \\
&= \frac{1}{V}\int_{\partial\overline{\mathcal{B}}_0} \mathbb{A} : \mathbb{I} + \mathbb{A} : \frac{\partial \widetilde{F}}{\partial \overline{F}}\ dV \\
&= \frac{1}{V}\int_{\partial\overline{\mathcal{B}}_0} \mathbb{A}\ dV + \frac{1}{V}\int_{\partial\overline{\mathcal{B}}_0} \mathbb{A} : \frac{\partial \widetilde{F}}{\partial \overline{F}}\ dV.
\end{aligned} \tag{4.9}$$

Hence, the computation of the macroscopic tangent moduli decomposes into the computation of the volumetric average of the microscopic tangent moduli and an additional term.

Note that the macroscopic stresses $\overline{P}$ as well as the macroscopic tangent moduli $\overline{\mathbb{A}}$ only have to be computed when the microscopic problems are converged, i.e., it can be assumed that the weak form of the balance of momentum on the microscopic level is fulfilled; see Equation (4.8). This can be utilized to derive a discrete version $\overline{\mathbb{A}}^h$ of an overall consistent macroscopic tangent modulus $\overline{\mathbb{A}}$ resulting from a finite element formulation; see [135, Ch. 3.2].

We assume that a finite element discretization into n finite elements T_i, $i = 1, \ldots, n$, of an RVE is given. Analogously, the macroscopic problem is discretized into m finite elements $\overline{T}_i$, $i = 1, \ldots, m$. The discrete version of the first term in Equation (4.9), i.e., the volumetric average of the tangent modulus on the microscopic level, writes

$$\frac{1}{V} \sum_{i=1}^{n} \int_{T_i} \mathbb{A}^h \, dV. \tag{4.10}$$

Here, $\mathbb{A}^h$ is the discrete version of the microscopic tangent modulus. To obtain the discrete version of the second term in Equation (4.9), we need the tangential element matrices

$$k_{T_i} = \int_{T_i} \mathbb{B}_{T_i}^T \mathbb{A}^h \mathbb{B}_{T_i} \, dV$$

as well as

$$l_{T_i} = \int_{T_i} \mathbb{A}^h \mathbb{B}_{T_i} \, dV.$$

Here, $\mathbb{B}_{T_i}$ contains the derivatives of the shape functions of the finite element T_i with respect to the reference coordinates. A standard assembly process of the tangential element matrices k_{T_i} yields the matrix K. In the same way, the matrix L is derived from the element matrices l_{T_i}. Note that the matrix K is identical to the matrix DK in [107, Ch. 5.1]. Furthermore, let us note that the matrix L has as much rows as the overall number of degrees of freedom in an RVE and the number of columns depends on the spatial dimensions of the problem. For two-dimensional problems, L has 4 columns and for three-dimensional problems, L has 9 columns.

Once the matrices L and K are computed, the discrete version of the second term in Equation (4.9) writes

$$-\frac{1}{V} L^T K^{-1} L, \tag{4.11}$$

which is derived from the weak form of the balance of momentum on the microscopic level. We refer to [135] for the derivation.

Finally, the overall consistent discrete macroscopic tangent modulus is obtained from Equations (4.10) and (4.11) and writes

$$\overline{\mathbb{A}}^h = \frac{1}{V} \int_{\mathcal{B}_0} \mathbb{A}^h \, dV - \frac{1}{V} L^T K^{-1} L. \tag{4.12}$$

4.2 Software Package FE2TI

For the simulations of the Nakajima test, we use our software package FE2TI. It was developed in the first phase of the EXASTEEL project (see [7]) and further developed in the second phase of the EXASTEEL project. The EXASTEEL project was part of the DFG priority programme 1648 *"Software for Exascale Computing"* (SPPEXA). In the past years, the FE2TI software scaled to some of the largest available HPC systems; see [74, 78, 79, 81]. Before we introduce the recent software features, we first describe the software package in general. Therefore, we follow the presentation in [107].

As already mentioned in the description of the FE^2 approach, we have to solve a boundary value problem on an RVE in each macroscopic integration point. The choice of the microscopic solver depends on the size of the resulting linearized system. For comparably small RVEs, the usage of a sparse direct solver such as MUMPS [1], UMFPACK [28], or MKL PARDISO [132] is recommended. They are provided via the PETSc interface. In this case, each RVE is computed on an individual core. If the microscopic problems lead to systems of equations that are too large to solve them efficiently with a sparse direct solver, one of our highly scalable parallel domain decomposition approaches (see Chapter 5) can be used to tackle the usually nonlinear problem. Such a domain decomposition method requires more than one core per RVE. Therefore, the computational cores have to be grouped into subsets where each subset is responsible for the parallel solution of one microscopic problem. As already mentioned in [107], we create different subsets by splitting the MPI_COMM_WORLD communicator into different subcommunicators using MPI_Comm_split. The microscopic tangent systems can be solved completely independently, i.e., there is no communication necessary between subcommunicators. The latter also holds for the averaging of the stresses and the overall consistent tangent modulus in one integration point. To provide the overall consistent macroscopic tangent modulus and all stresses to all necessary cores, collective communication is required.

Similar to the microscopic level, the choice of the solver for the linearized macroscopic problem also depends on its size. For small tangent systems, the same direct solvers as for the microscopic level are provided via the PETSc interface. If we use such a sparse direct solver, all compute cores redundantly solve the macroscopic problem. On one hand, this reduces communication since the macroscopic solution does not have to be distributed to other cores. On the other hand, all cores require the macroscopic mesh. Otherwise, if the macroscopic problem is too large to solve it efficiently with a sparse direct solver, we have recently integrated the Newton-Krylov BDDC approach (see Section 5.6.1)

for the parallel solution of the macroscopic problem. In this case, we introduce an additional subcommunicator to build up and subsequently solve the linearized system. Afterwards, collective communication is necessary to provide the solution to all cores. Again, we use MPI_Comm_split for the creation of the subcommunicator. It is also possible to use BoomerAMG [2,59] from the hypre package [37] for a parallel solution of the macroscopic problem, but it turned out that it does not perform well for the simulation of the Nakajima test. For completeness, it is also possible to use a parallel sparse direct solver, but this has not been tested so far.

Independently of the choice of the solver for the linearized problem on the macroscopic level, the assembly process of the macroscopic stiffness matrix and right-hand side is parallelized to save computation time. When using NK-BDDC, this is done automatically. In case of using a sparse direct solver, only a small number of cores is responsible for the assembly process. Since all cores have the macroscopic mesh, they can all participate in the assembly process without additional effort. The user can provide a real number between 0 and 1 which defines the fraction of overall cores that participate in the assembly process. It is usually about 1 % in our simulations. Once the assembly process is finished, the final stiffness matrix and right-hand side have to be provided to the other cores by collective communication.

Let us also give some general remarks regarding the software package FE2TI. As mentioned before, it is a C/C++ implementation based on PETSc and MPI. In all our simulations presented throughout this thesis, the macroscopic problem is discretized using triquadratic brick elements (Q_2) and the microscopic problem is discretized using piecewise quadratic tetrahedral elements (P_2) in an unstructured manner. Both meshes are generated using the open source software GMSH [49].

Throughout this thesis, the microscopic problem is always small enough to solve it efficiently with a sparse direct solver and we have used MKL PARDISO [132]. In our opinion, the simulation of the Nakajima test may be of interest for industrial application. Since most companies do not have access to the largest HPC systems, we found it reasonable to consider problem sizes which can be computed on mid-sized supercomputers. Therefore, in most simulations, also the macroscopic problem is chosen to be small enough to use a sparse direct solver.

So far, the software has only been used for the simulation of deformation processes of a DP steel, but it is not restricted to this case. The implementation of further microscopic material laws and the generation of other RVEs enables

the use of the software in many other fields of application. In our case, the microscopic material law is nonlinear, which also results in a nonlinear macroscopic problem, since the microscopic material properties are propagated to the macroscopic level by averaging. As a consequence, we have an iteration process on the macroscopic level. Of course, each macroscopic Newton iteration requires the solution of all microscopic problems, which are also nonlinear. Accordingly, we have an iteration process on the microscopic level in each macroscopic iteration. Furthermore, in simulations of real-world applications, the overall load is usually much too large to apply it in one step. Therefore, the total load is split into several small load steps, where each load step applies a fraction of the overall load and the solution of the current load step serves as initial value for the next load step. Let us note that each load step is associated with a pseudo time step. Since load stepping provides a solution whereas the application of the overall load cannot be solved, it is often used and it can be seen as a globalization strategy.

Considering the FE^2 approach in our software package FE2TI, the time to solution strongly depends on the number of load steps as well as on the number of macroscopic Newton iterations per load step. Furthermore, an individual macroscopic Newton iteration, again, depends on the time to solution of the microscopic problems. In the following, we introduce the software implementation in order to keep the number of load steps as well as the number of macroscopic Newton iterations as small as possible. The following presentations are based on the discussion in [87].

The number of load steps depends on the load step size, but its choice is critical. On one hand, too large load steps may lead to divergence of individual microscopic problems, which cause a termination of the simulation. On the other hand, too small load steps may increase the computing time. Therefore, we introduce a simple load step strategy to dynamically control the load step size; see Section 4.2.1.

The number of Newton iterations is affected by the choice of the initial value. The simplest choice of an initial value for the current load step is the converged solution of the previous load step; see Figure 4.1. It is also possible to introduce the predicted deformation of the current load step into the initial value; see Section 4.2.2. Choosing a better initial value reduces the number of Newton iterations per load step and the overall time to solution. In case of simulations of a tensile test using the FE2TI software package, the strategy of linear extrapolation (see Section 4.2.2) was successfully used; see [74].

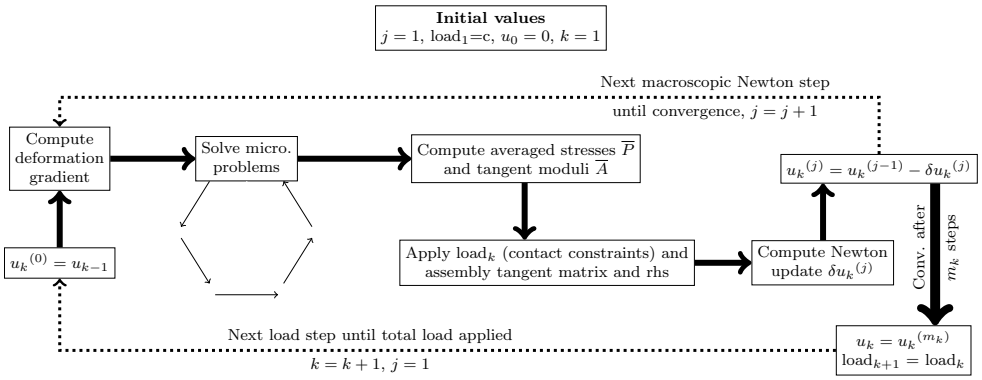


Figure 4.1: Schematic sequence of the FE^2 algorithm using a load step strategy.

Besides the software developments regarding the minimization of the number of load steps and macroscopic Newton iterations, we have also integrated a Checkpoint/Restart strategy, which is introduced in Section 4.2.3. Note that Section 4.2.3 is based on the presentations in [87]. The usage of a Checkpoint/Restart strategy reduces the consequences of hardware errors during the simulation. Additionally, it can be used to overcome specific wall time limits on HPC systems.

In addition, the simulation of the Nakajima test requires the incorporation of macroscopic contact constraints. Therefore, we have implemented a frictionless contact formulation using the penalty method into our software package FE2TI; see Section 4.2.4.

4.2.1 Dynamic Loadstepping

In our simulations, the mean time per load step strongly depends on the load step size, where a smaller load step size leads to a smaller mean timer per load step. However, a small load step size also leads to a large number of load steps to cover the overall load. It turns out that a larger load step size is advantageous in order to minimize the overall computing time; see Tables 4.1 and 4.2 as well as Tables 4.3 and 4.4. In simulations using a load step strategy, it is possible that a load step size works well in the beginning of the simulation, but may be too large in a later stage. In the context of FE2TI, a too large load step is associated with at least one diverging microscopic problem, which causes the termination of the simulation due to missing tangent moduli and stresses. Therefore, using a constant load step size is inappropriate. Instead, we use a dynamic load step strategy which enables the possibility to start with a comparable large load step size without causing the termination of the simulation in a later load step.

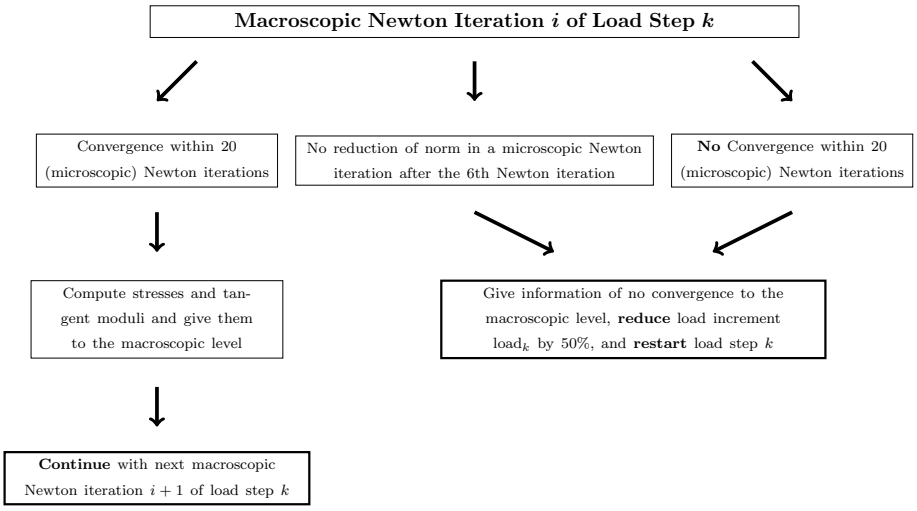


Figure 4.2: Impact of microscopic events on the load step size.

The strategy detects whether the load step size may be decreased, increased, or should remain constant; see Figure 4.4. For us, decreasing means to halve the load step size and increasing means to double it. The choice of the load step size depends on macroscopic as well as on microscopic information.

Based on microscopic information, the load step size may only be decreased. Whenever stagnation of a single microscopic problem is detected, this information is given to the macroscopic level, where the current load step is repeated with a halved load step size. Stagnation of a microscopic problem is detected when the norm of the solution of the current microscopic Newton iteration does not reduce compared to the previous one after the sixth microscopic Newton iteration or if convergence is not reached within 20 iterations; see Figure 4.2. Note that stagnation can occur in each microscopic problem of each macroscopic Newton iteration of a load step.

Based on macroscopic information, the load step size may be decreased or increased. Let us assume that convergence of a load step was reached within 20 macroscopic Newton iterations. The number of Newton iterations of the current load step has to be compared with the number of Newton iterations of the previous load step. If the number of the current load step is at most 50% of the number of the last load step, the load step size of the next load step is increased. Otherwise, the load step size remains constant. In case that convergence of a load step is not reached within 20 macroscopic Newton iterations, the load step size has to be decreased anyway. But if the norm of the solution of the 20th Newton iteration is close to the Newton tolerance, five more Newton iterations are spent. If convergence is reached within these five iterations, the load step

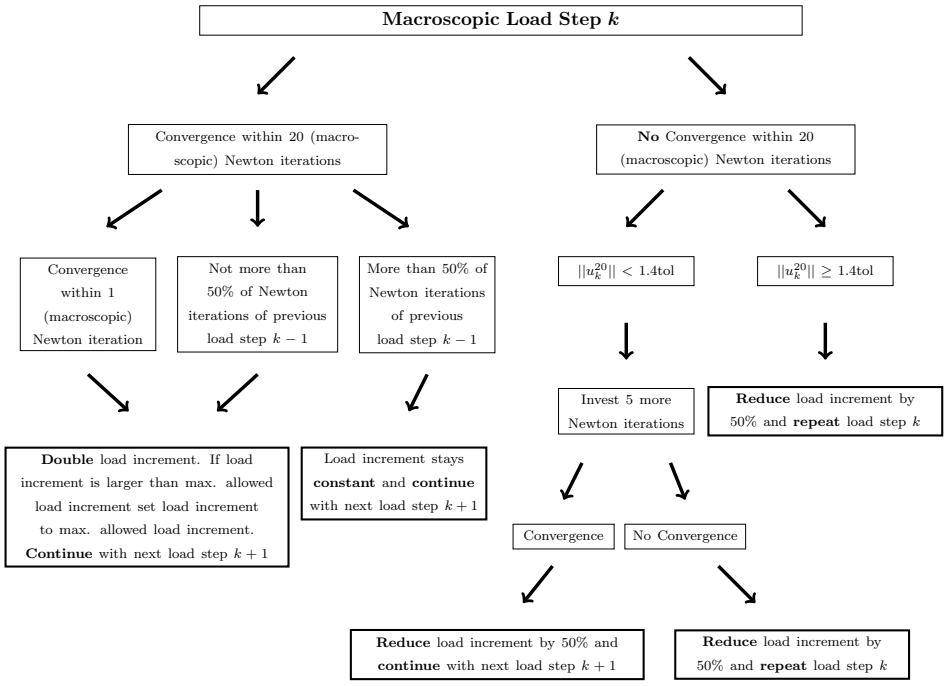


Figure 4.3: Impact of macroscopic events on the load step size.
This figure is similarly accepted and soon to be published; see [87, Fig. 6]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE.

size of the next load step has to be halved. Otherwise, the load step has to be repeated with a halved load step size. To prevent sticking to unnecessary small load step sizes, a load step size is increased whenever convergence of a load step is reached within one macroscopic Newton iteration. For an overview of all cases; see Figure 4.3.

To demonstrate the functionality of our dynamic load step strategy, we have compared different initial load step sizes with and without the usage of the dynamic load step strategy; see Tables 4.1 and 4.2. We have considered sample sheet geometries with parallel shaft widths of 50 mm (see Table 4.1) and 70 mm (see Table 4.2).

In both cases, we obtain similar results. For a comparably large load step size of 0.2 mm, the simulations using a constant load step size terminate within the second load step due to diverging microscopic problems. Instead, if we use our dynamic load step strategy, the simulation continues until the desired covered distance is reached. For the sample sheet with a parallel shaft width of 50 mm,

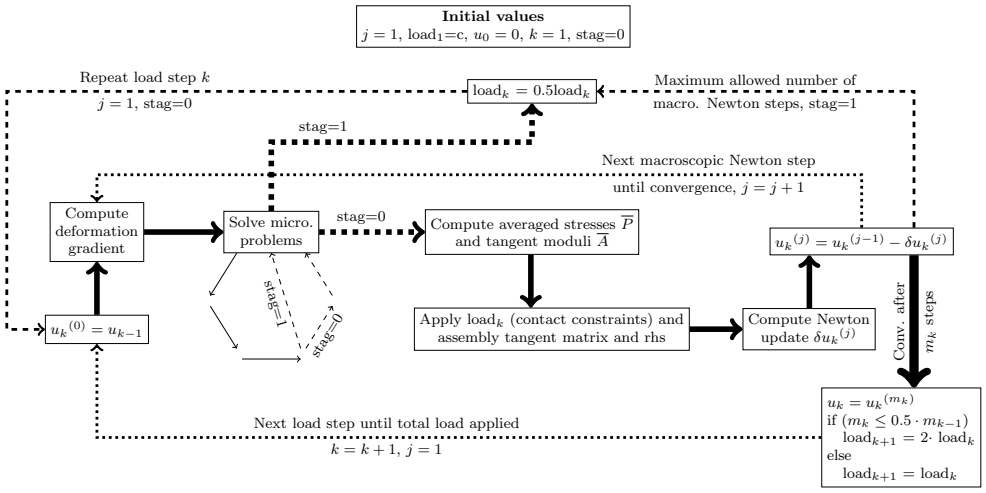


Figure 4.4: Schematic sequence of the FE^2 algorithm using a load step strategy with dynamic load step sizes depending on microscopic as well as macroscopic events (see Figures 4.2 and 4.3).

a load step size of 0.2 mm is suitable in a later stage. Therefore, the average load step size is close to 0.2 mm and the overall computing time is minimal compared to all other simulations; see Table 4.1. In contrast, after the load step size has to be decreased, the load step size remains constant (0.1 mm) for the rest of the simulation for the sample sheet with a parallel shaft width of 70 mm. Due to the overhead of the repeated load step, a constant load step size of 0.1 mm is slightly faster; see Table 4.2.

For comparably small load step sizes, the load step strategy detects that larger load step sizes are possible and raises the step size. Therefore, the absolute runtime can be significantly reduced.

Furthermore, an initial load step size of 0.1 mm seems to be optimal in some sense. If we use a linear extrapolation strategy (see Section 4.2.2), the dynamic load step strategy leaves the load step size unchanged, at least for the first 2 mm of tool movement; see also Tables 4.3 and 4.4. Of course, the load step size will be decreased if the rigid tool moves further. For example, for the sample sheet geometry with a parallel shaft width of 50 mm (see Table 3.1), we have required a minimum load step size of 0.00625 mm during the simulation of the Nakajima test until the critical value $\overline{W}_c$ was reached in finite element nodes on the top surface of the sample sheet.

Table 4.1: Comparison of some characteristic quantities for the first 2 mm covered by the rigid punch using different constant load step sizes as well as the dynamic load step strategy with different initial load step sizes; computed on JUWELS [70]; symmetric quarter of the sample sheet with a shaft width of 50 mm; two finite elements in thickness direction. We have used linear extrapolation and 2 MPI ranks per core. We consider the computation time as well as the number of macroscopic load steps and Newton iterations.

	Sample sheet 50 computed on JUWELS; 2 MPI ranks per core					
	Constant Load Step Size			**Dynamic Load Step Strategy**		
	Load 0.003125	**Load 0.1**	**Load 0.2**	**Load 0.003125**	**Load 0.1**	**Load 0.2**
Cov. Dist. Punch [mm]	2	2	term.	2	2	2
Load Steps	640	20	after	86	20	11
Newton Its.	970	130	one	328	130	91
∅ Load Step Size [mm]	0.003125	0.1	load	0.0233	0.1	0.18
∅ Newton Its. per Load Step	1.52	6.50	step	3.81	6.50	8.45
Runtime [s]	7 204.58	1 048.61		2 415.89	1 070.00	808.01
∅ Time per Load Step [s]	11.26	52.43		28.09	53.50	73.46
∅ Time per Newton It. [s]	7.43	8.07		7.37	8.23	8.88

4.2.2 Prediction of an Initial Value

For Newton-type methods, a good choice of the initial value is essential for superlinear convergence. If the initial value is close to the solution, only a few Newton iterations are required. Analogously to [74], we can use a linear interpolation polynomial to approximate the solution of the next load step.

Let us assume that we have just finished load step k and the accumulated load $\sum_{i=1}^{k-1} \bar{l}_i$ as well as the solution $\overline{u}_{k-1}$ of the previous load step $k-1$ are known, where $\bar{l}_j$ denotes the load increment of load step j. Furthermore, we assume that the load increment of load step $k+1$ is known. Therefore, the accumulated load $\sum_{i=1}^{k+1} \bar{l}_i$ is also known. Since every load step makes a small load increment, the accumulated loads of different load steps are obviously distinct. Accordingly, we obtain from, e.g., [127] that there exists a unique linear or constant interpolation polynomial

$$\mathfrak{p}_1(\bar{l}) = a \cdot \bar{l} + b, \ a, \ b \in \mathbb{R},$$

Table 4.2: Comparison of some characteristic quantities for the first 2 mm covered by the rigid punch using different constant load step sizes as well as the dynamic load step strategy with different initial load step sizes; computed on magnitUDE; symmetric quarter of a sample sheet with a shaft width of 70 mm; two finite elements in thickness direction. We have used linear extrapolation and 2 MPI ranks per core. We consider the computation time as well as the number of macroscopic load steps and Newton iterations.

	Sample sheet 70 computed on magnitUDE; 2 MPI ranks per core					
	Constant Load Step Size			**Dynamic Load Step Strategy**		
	Load 0.025	**Load 0.1**	**Load 0.2**	**Load 0.025**	**Load 0.1**	**Load 0.2**
Cov. Dist. Punch [mm]	2	2	term.	2	2	2
Load Steps	80	20	after	42	20	19
Newton Its.	344	136	one	216	136	138
∅ Load Step Size [mm]	0.025	0.1	load	0.048	0.1	0.105
∅ Newton Its. per Load Step	4.3	6.8	step	5.14	6.8	7.26
Runtime [s]	2 901.11	1 175.58		1 831.08	1 186.06	1 200.59
∅ Time per Load Step [s]	36.26	58.78		43.60	59.30	63.19
∅ Time per Newton It. [s]	8.43	8.64		8.48	8.72	8.70

with $\mathfrak{p}_1(\sum_{i=1}^{k-1} \bar{l}_i) = \overline{u}_{k-1}$ and $\mathfrak{p}_1(\sum_{i=1}^{k} \bar{l}_i) = \overline{u}_k$. The resulting interpolation polynomial writes

$$\mathfrak{p}_1(\bar{l}) = \overline{u}_{k-1} + \frac{\bar{l} - \sum_{i=1}^{k-1} \bar{l}_i}{\sum_{i=1}^{k} \bar{l}_i - \sum_{i=1}^{k-1} \bar{l}_i} \cdot (\overline{u}_k - \overline{u}_{k-1});$$

see [127, Eq. (1.4)]. This polynomial can be used to approximate the solution of the next load step $k+1$ by inserting the accumulated load $\sum_{i=1}^{k+1} \bar{l}_i$ into the polynomial. The approximated solution is subsequently used as initial value for load step $k+1$, i.e., the initial value $\overline{u}_{k+1}^{(0)}$ writes

$$\mathfrak{p}_1(\sum_{i=1}^{k+1} \bar{l}_i) =: \overline{u}_{k+1}^{(0)} = \overline{u}_{k-1} + \frac{\bar{l}_{k+1} + \bar{l}_k}{\bar{l}_k} \cdot (\overline{u}_k - \overline{u}_{k-1}). \tag{4.13}$$

In comparison to the presentation in [74], $\mathfrak{p}_1$ depends on the load increments of some load steps, since we do not consider constant load increments due to the dynamic load stepping introduced in Section 4.2.1. Let us note that all load

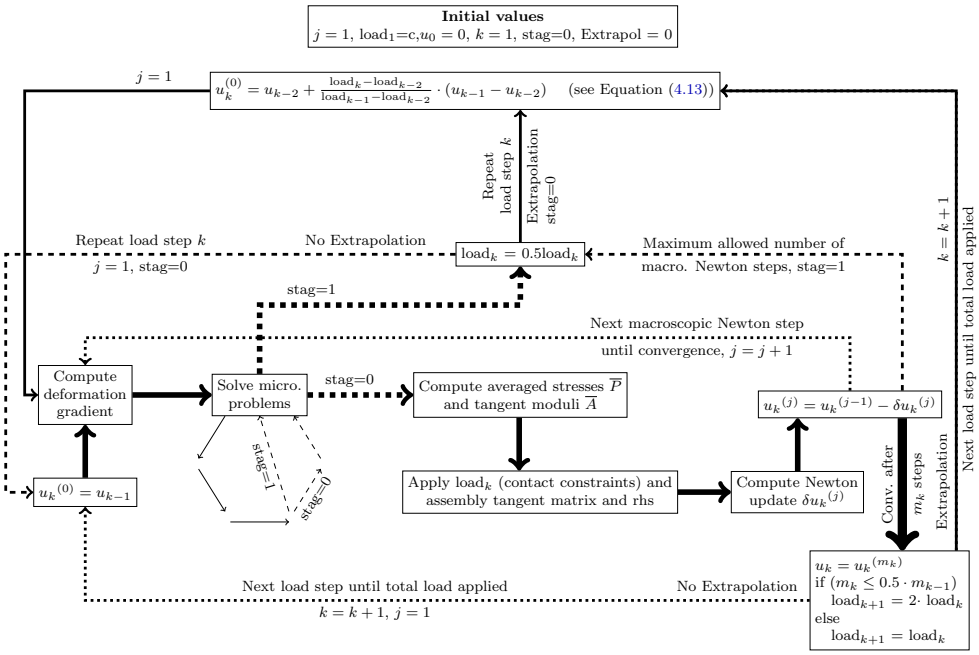


Figure 4.5: Schematic sequence of the FE^2 algorithm using a load step strategy with dynamic load step sizes and a linear extrapolation strategy for the computation of the initial value of the next load step. Instead of a linear extrapolation strategy, we can also use quadratic extrapolation; see Equation (4.13). The load step size depends on microscopic as well as macroscopic events (see Figures 4.2 and 4.3).

increments of load steps smaller than k vanish due to subtraction. The usage of a linear interpolation polynomial for the computation of an initial value for the next load step is referred to as linear extrapolation throughout this thesis.

As mentioned before, the macroscopic material behavior is nonlinear if the microscopic material model is nonlinear. Therefore, it might be advantageous to use a nonlinear interpolation polynomial instead of a linear one (see Equation (4.13)). Therefore, we also consider a polynomial $\mathfrak{p}_2$ of degree 2. In comparison to the linear polynomial, for which only the solutions and accumulated loads of the current and the previous load steps are required (see Equation (4.13)), we additionally need the solution and the accumulated load of the second last load step. Let us assume that we have just finished load step k and the solutions and accumulated loads of the previous load steps $k-1$ and $k-2$ as well as the load increment of load step $k+1$ are known. The interpolation polynomial $\mathfrak{p}_2(\bar{l}) = a \cdot \bar{l}^2 + b \cdot \bar{l} + c$, $a, b, c \in \mathbb{R}$, with $\mathfrak{p}_2(\sum_{i=1}^{k-2} \bar{l}_i) = \overline{u}_{k-2}$, $\mathfrak{p}_2(\sum_{i=1}^{k-1} \bar{l}_i) = \overline{u}_{k-1}$, and $\mathfrak{p}_2(\sum_{i=1}^{k} \bar{l}_i) = \overline{u}_k$ is unique, since the accumulated

loads of different load steps are distinct; see, e.g., [127]. We formulate $\mathfrak{p}_2$ in terms of the Lagrange polynomial, i.e., $\mathfrak{p}_2$ writes

$$\mathfrak{p}_2(\bar{l}) = \sum_{i=k-2}^{k} \overline{u}_i \cdot L_i(\bar{l}),$$

with

$$L_j(\bar{l}) = \prod_{\substack{m=k-2 \\ m\neq j}}^{k} \frac{\left(\bar{l} - (\sum_{s=1}^{m} \bar{l}_s)\right)}{\left((\sum_{s=1}^{j} \bar{l}_s) - (\sum_{s=1}^{m} \bar{l}_s)\right)};$$

see, e.g., [127]. Inserting the accumulated load after load step $k+1$ into $\mathfrak{p}_2$, i.e., we choose $\bar{l} = \sum_{i=1}^{k+1} \bar{l}_i$, we obtain the initial value $\overline{u}_{k+1}^{(0)}$ as

$$\begin{aligned} \mathfrak{p}_2(\sum_{i=1}^{k+1} \bar{l}_i) =: \overline{u}_{k+1}^{(0)} = {} & \frac{(\bar{l}_{k+1} + \bar{l}_k)\bar{l}_{k+1}}{\bar{l}_{k-1}(\bar{l}_k + \bar{l}_{k-1})} \overline{u}_{k-2} - \frac{(\bar{l}_{k+1} + \bar{l}_k + \bar{l}_{k-1})\bar{l}_{k+1}}{\bar{l}_{k-1}\bar{l}_k} \overline{u}_{k-1} \\ & + \frac{(\bar{l}_{k+1} + \bar{l}_k + \bar{l}_{k-1})(\bar{l}_{k+1} + \bar{l}_k)}{(\bar{l}_k + \bar{l}_{k-1})\bar{l}_k} \overline{u}_k. \end{aligned} \tag{4.14}$$

Similar to the linear polynomial, load increments of load steps smaller than $k-1$ vanish due to subtraction. Throughout this thesis, we refer to the usage of an initial value computed with an interpolation polynomial of degree 2 as quadratic extrapolation.

The algorithmic scheme using extrapolation for the computation of an improved initial value is shown in Figure 4.5. In order to save some space, only the case of linear extrapolation is shown, but for quadratic extrapolation, only the formula for calculating the initial value has to be exchanged (see Equations (4.13) and (4.14)). Since the quadratic extrapolation has been implemented very recently, most simulation results including all simulation for the derivation of the FLD are performed using the linear extrapolation strategy.

Let us compare the different extrapolation strategies. Therefore, we consider the first 4 mm of tool movement for a symmetric quarter of a sample sheet with a parallel shaft width of 40 mm (see Table 4.3) as well as for a complete sample sheet with a parallel shaft width of 90 mm (see Table 4.4). In the latter case, we exclusively consider finite elements that do not belong to the clamped area between the blank holder and die; see Sections 2.4 and 2.5. For the simulation of the complete sample sheet, we use a finite element discretization with one finite element in thickness direction to reduce the computational costs.

As a baseline for our comparison, we consider a simulation without using any extrapolation but taking into account our dynamic load step strategy intro-

duced in Section 4.2.1. Note that quadratic extrapolation cannot be activated before convergence of the third load step, because it requires information from the previous two load steps. In comparison, linear extrapolation can already be activated one step earlier. We have activated both extrapolation strategies as soon as possible. To provide a fair comparison, we have also activated linear extrapolation at the same time as quadratic extrapolation. It would be also possible to perform one step of linear extrapolation and thereafter activate quadratic extrapolation, but this has not been considered so far.

We obtain similar results for the simulation of both sample sheets; see Tables 4.3 and 4.4. Even if we use a constant load step size, the number of overall macroscopic Newton iterations can be significantly reduced. As one can see in Tables 4.3 and 4.4, the usage of the quadratic extrapolation seems to be more efficient since it needs less overall macroscopic Newton iterations and, therefore, saves more than 10% of computing time compared to linear extrapolation.

As mentioned in the previous section, the load step size remains constant for a comparably small covered distance of the rigid punch if we use an initial load step size of 0.1 mm in combination with a linear extrapolation strategy. This is different in case of quadratic extrapolation. After activating quadratic extrapolation, the number of macroscopic Newton iterations drops such that less than 50% of the previous number of macroscopic Newton iterations are needed. Thus, the load step size is increased. After a while, the load step size again has to be decreased due to diverging microscopic problems. Nonetheless, the average load step size is larger and, therefore, less load steps are necessary compared to linear extrapolation. It is striking that the average number of Newton steps per load step is smaller compared to linear extrapolation, although a larger average load is used. This shows once again that quadratic extrapolation can be worthwhile. Finally, the usage of quadratic extrapolation using a dynamic load step strategy can reduce the computing time by more than 50% compared to the simulation without any extrapolation strategy; see Table 4.4.

The drawback of the quadratic extrapolation is an additional storage of a solution vector since we require the solution of the last two load steps. Depending on the HPC system, it might occur that we are memory bounded, especially for large macroscopic problems. If this is the case, the quadratic extrapolation strategy cannot be used. Furthermore, the additional solution vector also has to be integrated into our checkpoint (see Section 4.2.3), which also takes some extra time for large macroscopic problems.

Table 4.3: Comparison of linear and quadratic extrapolation for the first 4 mm covered by the rigid punch with and without using a dynamic load step strategy (see Section 4.2.1); initial load step size of 0.1 mm; computed on magnitUDE; symmetric quarter of the sample sheet with a shaft width of 40 mm; two MPI ranks per core; two finite elements in thickness direction.

	Sample sheet 40 computed on magnitUDE; 2 MPI ranks per core						
	Const. Load Step Size			**Dynamic Load Step Strategy**			
	Lin. Extra.	**Lin. Extra.**	**Quad. Extra.**	**No Extra.**	**Lin. Extra.**	**Lin. Extra.**	**Quad. Extra.**
Cov. Dist. Punch [mm]	4	4	4	4	4	4	4
1. Load Step Using Extra.	3	4	4	–	3	4	4
Load Steps	40	40	40	45	40	40	33
Newton Its.	310	312	273	562	310	312	245
∅ Load Step Size [mm]	0.1	0.1	0.1	0.089	0.1	0.1	0.12
∅ Newt. Its. per Load Step	7.75	7.8	6.83	12.49	7.75	7.8	7.42
Runtime [s]	2 435.86	2 451.4	2 158.09	4 409.67	2 414.47	2 435.57	1 943.37
∅ Time per Load Step [s]	60.90	61.28	53.95	97.99	60.36	60.89	58.59
∅ Time per Newt. It. [s]	7.86	7.86	7.91	7.85	7.79	7.81	7.93

Let us note that the use of extrapolation changes the role of the contact constraints. While the contact constraints exclusively cause the deformations without using an extrapolation strategy, they have a slightly different task if an extrapolation strategy is used. No matter which polynomial we use to calculate an initial value, in both cases we obtain an approximate solution of the next load step. As a consequence, the initial value already contains deformations that have not been driven by the contact constraints. Therefore, the contact constraints have to check the deformations contained in the initial value and adjust them if necessary.

4.2.3 Checkpoint/Restart

The virtual derivation of an FLD and its corresponding FLC requires the simulation of the Nakajima test with different sample sheets. For each sample sheet, the simulation has to be performed until a failure zone occurs, which takes more than 14 hours (see Table 3.1), even if the full supercomputer is available. Longer runtimes automatically increase the risk of hardware failures during the simu-

Table 4.4: Comparison of linear and quadratic extrapolation for the first 4 mm covered by the rigid punch with and without using a dynamic load step strategy (see Section 4.2.1); initial load step size of 0.1 mm; computed on magnitUDE; no utilization of symmetry for a sample sheet with a shaft width of 90 mm; two MPI ranks per core; one finite element in thickness direction.

	Sample sheet 90 computed on magnitUDE; 2 MPI ranks per core						
	Const. Load Step Size			**Dynamic Load Step Strategy**			
	Lin. Extra.	**Lin. Extra.**	**Quad. Extra.**	**No Extra.**	**Lin. Extra.**	**Lin. Extra.**	**Quad. Extra.**
Cov. Dist. Punch [mm]	4	4	4	4	4	4	4
1. Load Step Using Extra.	3	4	4	–	3	4	4
Load Steps	40	40	40	40	40	40	34
Newton Its.	332	334	286	445	332	334	262
∅ Load Step Size [mm]	0.1	0.1	0.1	0.1	0.1	0.1	0.12
∅ Newt. Its. per Load Step	8.3	8.35	7.15	11.13	8.3	8.35	7.71
Runtime [s]	2 684.55	2 688.70	2 326.75	3 623.05	2 672.15	2 683.78	2 157.91
∅ Time per Load Step [s]	67.11	67.22	58.17	90.58	66.80	67.09	63.47
∅ Time per Newt. It. [s]	8.09	8.05	8.14	8.14	8.05	8.04	8.24

lations. To reduce the consequences of such hardware failures on the one hand and to overcome specific wall time limits of supercomputing systems on the other hand, we equipped our FE2TI software with a Checkpoint/Restart (CR) strategy. Therefore we integrated the CRAFT library (**C**heckpoint/**R**estart and **A**utomatic **F**ault **T**olerance) [136], which was developed in the second phase of the SPPEXA project ESSEX. Even if the library enables many more possibilities such as automatic fault tolerance, we only use it for Checkpoint/Restart so far. In CRAFT, we can choose between synchronous and asynchronous Checkpoint/Restart. By default, asynchronous Checkpoint/Restart creates local copies of the checkpointing data, which might be large for us. Furthermore, we can only write a single checkpoint when using the asynchronous variant. Accordingly, we choose synchronous Checkpoint/Restart in our simulations.

Let us note that we use different checkpoint objects for macroscopic values and microscopic values. Furthermore, we have a third checkpoint object, which writes a checkpoint immediately before termination of the simulation, i.e, if the rigid punch has reached the desired covered distance or if too much consecutive load steps with a very small load step size were performed. The latter checkpoint

object can be used to increase the contact stiffness $\overline{\varepsilon}_N$ for the final solution in an additional simulation. To prevent us from simulating load steps twice in a restart run, we recently implemented the option to write a checkpoint and to stop the simulation if the wall time is almost reached.

So far, we choose a simple checkpoint interval depending on a specific number of load steps, which can be chosen by the user. With this strategy, the time interval between two checkpoints strongly depends on the load step size, since the mean time per load step depends on the load step size; see Tables 4.1 to 4.4. Hence, the time between two checkpoints is much smaller if the load step size is small, which is usually the case at the end of the simulation. One could also think about a checkpoint interval based on a fixed covered distance of the rigid punch. Here, the time interval between two checkpoints may be too large if a small load step size is used. As an improvement, the checkpoint interval can also be based on a wall time interval which additionally could depend on the expected runtime and the mean time of hardware failure on the used HPC system; see [27] for different checkpoint intervals based on this idea.

4.2.4 Frictionless Contact Between a Rigid Tool and a Deformable Body

In the Nakajima test, the sample sheet comes into contact with several rigid tools, namely the hemispherical punch, the blank holder, and the die. Hence, the simulation of the Nakajima test requires the incorporation of macroscopic contact into the FE2TI software package. As mentioned in Chapter 2, friction between the rigid punch and the sample sheet has to be avoided as much as possible by using a sufficient lubrication system. Since the deformation process is completely driven by the frictionless contact between the hemispherical punch and the sample sheet and is only restricted through contact with the die and the blank holder, we consider a formulation for frictionless contact between a rigid tool and a deformable body. In the following, the deformable body is denoted by $\overline{\mathcal{B}}$ and the different rigid tools by $\overline{\mathcal{T}}_i$, $i = 1, 2, 3$. The hemispherical punch is referred to as $\overline{\mathcal{T}}_1$, the blank holder is denoted by $\overline{\mathcal{T}}_2$, and the die is denoted by $\overline{\mathcal{T}}_3$. For simplicity, we consider contact between the deformable body $\overline{\mathcal{B}}$ and an arbitrary rigid tool $\overline{\mathcal{T}}$ in the following. The following discussion is based on the presentations in [87].

In contact formulations, one contact partner is assumed to be the master body and the other body is referred to as slave body [102, 151]. Only points of the contact surface of the slave body are allowed to penetrate into the master body. In the case of contact between a deformable body and a rigid tool, as

considered here, the amount of penetration can be computed using the rigid body as master body or as slave body [102]. In our implementation, we follow the recommendation from [151, Rem. 4.3] to use the rigid body contact surface as master surface. The resulting contact contributions to the stiffness matrix and the right-hand side are computed in the coordinate system of the deformable body, regardless of the choice of the master body.

Let us assume that only points of one surface of the deformable body $\overline{\mathcal{B}}$ can come into contact with the rigid tool $\overline{\mathcal{T}}$ and that the surface is denoted by $\overline{\Gamma}_{\overline{\mathcal{B}}}$. In each iteration, we have to check for each finite element node $x_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}$ whether it penetrates into the master body $\overline{\mathcal{T}}$ or not. Therefore, we have to determine for each $x_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}$ the related point on the rigid tool surface $\overline{\Gamma}_{\overline{\mathcal{T}}}$, which has minimum distance to $x_{\overline{\mathcal{B}}}$, i.e., we are looking for

$$\overline{x}_{\overline{\mathcal{T}}}^{\min} := \min_{\overline{x}_{\overline{\mathcal{T}}} \in \overline{\Gamma}_{\overline{\mathcal{T}}}} ||\overline{x}_{\overline{\mathcal{B}}} - \overline{x}_{\overline{\mathcal{T}}}||.$$

Once we have found the related minimum distance point $\overline{x}_{\overline{\mathcal{T}}}^{\min}$, we have to compute the outward normal vector $\overline{n}_{\overline{\mathcal{T}}}^{\min}$ at this point. This can be done as usual as cross product of two tangent vectors which are orthogonal to each other. When using the penalty method for the incorporation of the contact constraints, the outward normal vector can also be computed as

$$\overline{n}_{\overline{\mathcal{T}}}^{\min} = \frac{\overline{x}_{\overline{\mathcal{B}}} - \overline{x}_{\overline{\mathcal{T}}}}{||\overline{x}_{\overline{\mathcal{B}}} - \overline{x}_{\overline{\mathcal{T}}}||}. \tag{4.15}$$

For the computation of the outward normal vectors of the different rigid tools in the Nakajima test, we refer to Section 4.2.4.3.

Since in reality one solid cannot penetrate into another, we want to include this condition into the numerical simulation. The mathematical formulation of the non-penetration condition is

$$\overline{g}_{NP}(\overline{x}_{\overline{\mathcal{B}}}) = \left(\overline{x}_{\overline{\mathcal{B}}} - \overline{x}_{\overline{\mathcal{T}}}^{\min}\right)^T \cdot \overline{n}_{\overline{\mathcal{T}}}^{\min} \geq 0, \ \overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}. \tag{4.16}$$

In conjunction with the penalty method, we have to compute the amount of penetration for all finite element nodes that violate Equation (4.16). Therefore, we denote the set of active contact constraints as

$$\overline{\Gamma}_c := \left\{ \overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}} \;\middle|\; \overline{g}_{NP}(\overline{x}_{\overline{\mathcal{B}}}) < 0 \right\};$$

see red squares in Figure 4.6. For all finite element nodes $\overline{x}_c \in \overline{\Gamma}_c$, the outward normal at the corresponding minimum distance point writes $\overline{n}_{\overline{\mathcal{T}}}^{\min} = -\frac{\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}}{||\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}||}$

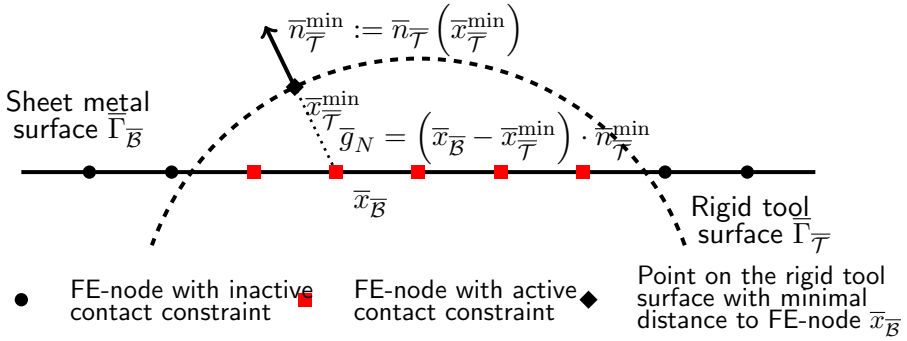


Figure 4.6: Illustration for the determination of active contact nodes and the amount of penetration.
This figure is accepted and soon to be published; see [87, Fig. 5]; EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing; Software for Exascale Computing - SPPEXA 2016-2019; Springer LNSCE.

and the amount of penetration $\overline{g}_N$ can be computed as

$$\begin{aligned}
\overline{g}_N(\overline{x}_c) = -||\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}|| &= -\frac{||\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}||^2}{||\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}||} \\
&= -\left(\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}\right)^T \cdot \frac{\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}}{||\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}||} \\
&= \left(\overline{x}_c - \overline{x}_{\overline{\mathcal{T}}}^{\min}\right)^T \cdot \overline{n}_{\overline{\mathcal{T}}}^{\min} \\
&= \overline{g}_{NP}(\overline{x}_c). \qquad (4.17)
\end{aligned}$$

The amount of penetration is set to zero for the remaining finite element nodes $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}} \setminus \overline{\Gamma}_c$ on the contact surface of $\overline{\mathcal{B}}$, i.e., $\overline{g}_N(x) = 0 \ \forall x \in \overline{\Gamma}_{\overline{\mathcal{B}}} \setminus \overline{\Gamma}_c$. For an illustration presenting the basic ideas discussed in this section; see Figure 4.6.

The rigid tool $\overline{\mathcal{T}}$ is not discretized by finite elements, but its surface is characterized by an analytical function. This simplifies the computation of the related minimum distance point, and thus the calculation of the outward normal vector and of the amount of penetration; see Section 4.2.4.3. For a detailed description of contact kinematics between two deformable bodies, we refer to [151, Sec 4.1].

As usual in standard finite element simulations of continuum mechanical problems, we are interested in the minimization of an energy functional $\widetilde{\Pi}$. Due to the non-penetration condition (cf. Equation (4.16)), we have to consider

a constrained optimization problem, which can be solved by, e.g., the penalty method.

4.2.4.1 Penalty Method

The solution of constrained optimization problems is required in many applications. For example, in the simulation of contact driven simulations it must be ensured that the bodies in contact do not penetrate each other. Mathematically, this condition can be formulated as an inequality constraint (see Equation (4.16)) and it has to be taken into account when minimizing the resulting energy functional.

A prominent approach for the approximative solution of constrained minimization problems is the quadratic penalty method; see [122, Ch. 17.1] and [113, Ch. 13]. The general idea of penalty methods is to approximate the solution of the constraint minimization problem by solving an unconstrained minimization problem. The objective function of the unconstrained minimization problem results from the objective function of the original problem and additional terms which penalize the violation of the constraint equations. Small violations of a specific constraint are less penalized than large violations.

In detail, the solution of minimizing a function

$$\begin{aligned} f : \mathbb{R}^n &\to \mathbb{R} \\ x &\mapsto f(x) \end{aligned}$$

under some constraint equations $c_i(x) \geq 0,\ i \in \mathcal{I} \subseteq \{1, \ldots, n\}$, is approximated by solving an unconstrained minimization problem of a function

$$\begin{aligned} g : \mathbb{R}^n &\to \mathbb{R} \\ x &\mapsto f(x) + \frac{\overline{\varepsilon}_N}{2} \sum_{i \in \mathcal{I}} \max(-c_i(x)^2, 0). \end{aligned}$$

Here, $\overline{\varepsilon}_N > 0$ is a penalty parameter, which is often denoted by c in the literature; see, e.g., [11, 113]. In the context of contact between a deformable body and a rigid tool, we have an individual constraint equation for each possible contact node. Let us define the **active set** $\mathcal{A}$ of contact constraints as

$$\mathcal{A} := \left\{ i \in \mathcal{I} \;\middle|\; c_i(x) < 0 \right\} \subseteq \mathcal{I}. \tag{4.18}$$

This definition differs from the notation in the literature, where an active set contains all indices $i \in \mathcal{I}$ with $c_i(x) = 0$ (see, e.g., [113, 122]). However, using

Equation (4.18), we can rewrite g to

$$g(x) = f(x) + \frac{\overline{\varepsilon}_N}{2} \sum_{i \in \mathcal{A}} c_i(x)^2,$$

since constraints with $c_i(x) = 0$ do not contribute to the objective function $g(x)$.

Let us assume that both, the constrained as well as the unconstrained minimization problems have a solution. Then, the penalty parameter $\overline{\varepsilon}_N$ implicitly determines the accuracy of the approximative solution. For small penalty parameters, violation of the constraint equations is less penalized and hence, the approximative solution is poor. In contrast, for large penalty parameters, violation of the constraints generate high costs, even for small violations, i.e., the solution x_g of $g(x)$ is close to the solution x^* of $f(x)$. In practice, a sequence $\left\{x_g^k\right\}$ of minima of $g(x, \overline{\varepsilon}_N^k)$ is generated with an increasing penalty parameter tending to infinity, i.e., $\lim_{k\to\infty} \overline{\varepsilon}_N^k = \infty$ and $\overline{\varepsilon}_N^k > \overline{\varepsilon}_N^{k-1}$; see [113]. It can be shown that the sequence $\left\{x_g^k\right\}$, generated by the penalty method, converges to the solution x^* of the constraint minimization problem of f; see [113, Ch. 13, p. 400].

In [151], it is noted that $\overline{\varepsilon}_N \to \infty$ yields the Lagrange multiplier method, which can also be used for the solution of constrained minimization problems; see [113, 122].

In contact simulations, we do not compute a sequence of rising penalty parameters for each load step but choose a constant penalty parameter throughout the simulation process. However, the penalty parameter might be increased during the simulation or at its end.

A disadvantage of the penalty approach is the fact that for large penalty parameters the Hessian matrix becomes ill-conditioned. For $\overline{\varepsilon}_N^k \to \infty$, $|\mathcal{A}|$ eigenvalues of the Hessian matrix also tend to infinity; see [113, p. 406ff] for further details. Note that the convergence rate of Newton's method is not affected by the structure of the eigenvalues of the Hessian, but the inverse of the ill-conditioned Hessian has to be computed carefully; see [113, Ch. 13.4].

For further details, we refer to [122, Ch. 17], [113, Ch. 13], and [11, Ch. 2.1].

4.2.4.2 Contact Formulation Using the Penalty Method

The treatment of frictionless contact as a constrained optimization problem follows directly from the contact formulation. It is obvious that there are no contact stresses $\overline{P}_N$ in the contact interface if $\overline{g}_N > 0$ holds for all $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}$, i.e., $\overline{g}_N > 0$ implies $\overline{P}_N = 0$. In case that the bodies are in contact, $\overline{g}_N = 0$ has

to hold and the contact stress $\overline{P}_N$ is not zero. Following [151], adhesive stresses are forbidden in the contact interface. Hence, in case of contact, $\overline{P}_N < 0$ holds. The latter explanations can be combined to

$$\overline{g}_N \geq 0, \qquad \overline{P}_N \leq 0, \qquad \overline{g}_N \cdot \overline{P}_N = 0. \tag{4.19}$$

These conditions are called Hertz-Signorini-Moreau conditions (see [151]) and are equivalent to the Karush-Kuhn-Tucker (KKT) conditions (see, e.g., [11,113, 122]).

There are many different methods to incorporate contact constraints into finite element formulations; see [151, Sec. 6.3]. Besides the penalty method, the method of Lagrange multipliers is the most commonly used method. While the number of unknowns does not increase in the case of the penalty method, the contact constraints are only resolved approximately. The amount of allowed penetration depends on the choice of the penalty parameter $\overline{\varepsilon}_N$, which can be interpreted as the stiffness of a spring that is placed in the contact interface of the deformable body and the rigid tool [151, Sec. 2.1.3]. For a suggestion of the choice of the penalty parameter, we refer to [151, Rem. 10.2].

Let us note that the equations describing the behavior of the bodies coming into contact are not affected by the incorporation of the contact constraints [151]. Using the quadratic penalty method, we have to add the additional term

$$\widetilde{\Pi}_P = \int_{\overline{\Gamma}_c} \frac{1}{2} \cdot \overline{\varepsilon}_N \cdot \overline{g}_N^2 \, \mathrm{dA}$$

to the energy functional $\widetilde{\Pi}$ [151, Sec. 6.3]. Hence, all active finite element nodes have an impact to the overall energy functional. Instead of using the notation of active contact constraints, other authors like Konyukhov and Schweizerhof introduce the Heaviside function; see, e.g., [101,102]. The Heaviside function is also called unit step function and can be found in, e.g., [71, Sec. 1.1] or [29, Eq. 7.94].

In FE2TI, we solve the resulting weak formulation including the contact constraints iteratively by using Newton's method. Hence, also the contact constraints have to be linearized, which leads to an additional contact part in the resulting stiffness matrix. Moreover, we also obtain an additional contact part in the right-hand side. Following the representation in [67], the impact of a single contact constraint to the stiffness matrix of a single finite element writes

$$K = K_{\mathrm{main}} + K_{\mathrm{rot}} + K_{\mathrm{curv}}.$$

The different parts of K in the latter equation are derived by numerical integration over the contact surface of the finite element. We obtain for an active contact constraint

$$
\begin{aligned}
K_{\text{main}} &= \overline{\varepsilon}_N \cdot \sum_{q=1}^{n_{gp}} A^T(\overline{n} \otimes \overline{n}) A \cdot w_q \cdot b, \\
K_{\text{rot}} &= -\overline{\varepsilon}_N \cdot \overline{g}_N \sum_{q=1}^{n_{gp}} w_q \cdot b \sum_{i=1}^{2} \sum_{j=1}^{2} A_{,j}^T \cdot a^{ij} (\overline{n} \otimes \overline{a}_{,i}) A + A^T (\overline{a}_{,i} \otimes \overline{n}) \cdot a^{ji} A_{,j}, \\
K_{\text{curv}} &= -\overline{\varepsilon}_N \cdot \overline{g}_N \sum_{q=1}^{n_{gp}} w_q \cdot b \sum_{i=1}^{2} \sum_{j=1}^{2} A^T (\overline{a}_{,i} \otimes \overline{a}_{,j}) h^{ij} A.
\end{aligned}
$$

Here, n_{gp} is the number of quadrature points and w_q are its corresponding weights. The value $b = \sqrt{\det a_{ij}}$ is the square root of the determinant of the metric tensor written in a covariant basis, a^{ij} are the components of the metric tensor in a contravariant basis, and h^{ij} is the curvature tensor in the contravariant basis. The vector $\overline{a}_{,i}$, $i = 1, 2$, is the tangent vector in direction i at the quadrature point and the matrices A and $A_{,j}$ write

$$
A = - \begin{bmatrix} N^1 & 0 & 0 & \dots & N^{n_{gp}} & 0 & 0 \\ 0 & N^1 & 0 & \dots & 0 & N^{n_{gp}} & 0 \\ 0 & 0 & N^1 & \dots & 0 & 0 & N^{n_{gp}} \end{bmatrix},
$$

$$
A_{,j} = - \begin{bmatrix} N_j^1 & 0 & 0 & \dots & N_j^{n_{gp}} & 0 & 0 \\ 0 & N_j^1 & 0 & \dots & 0 & N_j^{n_{gp}} & 0 \\ 0 & 0 & N_j^1 & \dots & 0 & 0 & N_j^{n_{gp}} \end{bmatrix},
$$

where N^i are the basis functions of the finite element surfaces and N_j^i, $j = 1, 2$, are their partial derivatives; see [67].

In our implementation, we use Q_2 finite elements. Hence, the contact surface of a finite element has 9 quadrature points. Furthermore, we check for penetration at the finite element nodes. Therefore, we consider the Lobatto integration with its weights $\frac{1}{9}$ for the corners, $\frac{4}{9}$ for midpoints of edges, and $\frac{16}{9}$ for the midpoint of the surface. Note that we use only the main part of the stiffness matrix in our implementation since the rotational and the curved parts are small due to the multiplication with $\overline{g}_N$.

For a detailed derivation of the linearized terms as well as for the definition of the metric tensor and the curvature tensor we refer to different works of Konyhukov and Schweizerhof [67,101,102] as well as to [151, Sec. 6.5] and [151, App. B].

4.2.4.3 Contact With Rigid Tools in the Nakajima Test

As mentioned before, the computation of the corresponding projection point on the rigid surface as well as its associated outward normal vector simplifies if the rigid surface is characterized by an analytical function. In the Nakajima test, the sample sheet is in contact with the hemispherical punch, the blank holder, and the die. While the blank holder is only a horizontal plane, the die consists of three different parts, namely a horizontal flat part, a vertical flat part, and a curved part in between. Therefore, we have to distinguish between three different cases, which are contact with the rigid punch, contact with a flat part, and contact with the curved part of the die; see Figure 2.1 and the different radii in Figure 2.4.

Contact with the Rigid Punch Let us first consider contact between the sheet metal and the forming tool $\overline{\mathcal{T}}_1$. We assume that the radius $\overline{r}_s$ and the initial position of the center $\overline{c}_s = \left[\overline{c}_s^1, \overline{c}_s^2, \overline{c}_s^3\right] \in \mathbb{R}^3$ of the hemispherical punch are known. Then, the surface of the hemispherical punch can be characterized by the function $||\overline{x} - \overline{c}_s|| = \overline{r}_s,\ \overline{x} \in \mathbb{R}^3$. Note that the third coordinate $\overline{c}_s^3$ of the center of the hemispherical punch changes in every load step. With the analytical function describing the surface of the rigid punch $\overline{\mathcal{T}}_1$, we can formulate an alternative non-penetration condition

$$\hat{g}_{NP}^{\overline{\mathcal{T}}_1}(\overline{x}_{\overline{\mathcal{B}}}) = ||\overline{x}_{\overline{\mathcal{B}}} - \overline{c}_s|| - \overline{r}_s \geq 0,\ \overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}.$$

For any finite element node $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}$, the outward normal direction $\overline{n}_{\overline{\mathcal{T}}_1}^{\min}$ at the related minimum distance point $\overline{x}_{\overline{\mathcal{T}}_1}^{\min}$ of the upper half of the sphere is defined through the direction from the center $\overline{c}_s$ to $\overline{x}_{\overline{\mathcal{B}}}$, which yields

$$\overline{n}_{\overline{\mathcal{T}}_1}^{\min} = \frac{\overline{x}_{\overline{\mathcal{B}}} - \overline{c}_s}{||\overline{x}_{\overline{\mathcal{B}}} - \overline{c}_s||}.$$

The related minimum distance point $\overline{x}_{\overline{\mathcal{T}}_1}^{\min}$ to $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}$ can be derived in closed form as $\overline{x}_{\overline{\mathcal{T}}_1}^{\min} = \overline{c}_s + \overline{n}_{\overline{\mathcal{T}}_1}^{\min} \cdot \overline{r}_s$.

Contact with a Flat Rigid Surface Parallel to a Coordinate Plane Things become even more simple for a flat contact surface parallel to one of the three coordinate planes. We consider a rigid body $\overline{\mathcal{T}} \in \mathbb{R}^3$ with a flat contact surface $\overline{\Gamma}_{\overline{\mathcal{T}}}^i,\ i \in \{1,2,3\}$, which is parallel to one of the coordinate planes. The contact surface $\overline{\Gamma}_{\overline{\mathcal{T}}}^i$ is defined by all points $\overline{p} = \left[\overline{p}^1, \overline{p}^2, \overline{p}^3\right] \in \overline{\mathcal{T}}$, where a single component $\overline{p}^i,\ i \in \{1,2,3\}$, is a constant value $\overline{c}_f \in \mathbb{R}$, i.e., $\overline{p}^i = \overline{c}_f$ for one $i \in \{1,2,3\}$. The corresponding outward normal is $\pm e_i$, where e_i

is the i-th unit vector. The sign depends on the position of the deformable body $\overline{\mathcal{B}}$ in comparison to the rigid surface $\overline{\Gamma}_{\overline{\mathcal{T}}}$. Subsequently, for any finite element node $\overline{x}_{\overline{\mathcal{B}}} = \left[\overline{x}^1_{\overline{\mathcal{B}}}, \overline{x}^2_{\overline{\mathcal{B}}}, \overline{x}^3_{\overline{\mathcal{B}}}\right] \in \overline{\mathcal{B}}$, the related minimum distance point $\overline{x}^{\min}_f = \left[\overline{x}^{\min,1}_f, \overline{x}^{\min,2}_f, \overline{x}^{\min,3}_f\right] \in \overline{\Gamma}^i_{\overline{\mathcal{T}}}$ is defined by $\overline{x}^{\min,i}_f = \overline{c}_f$, $\overline{x}^{\min,j}_f = \overline{x}^j_{\overline{\mathcal{B}}}$, $i \in \{1,2,3\}$, $j \neq i$. Finally, the alternative non-penetration condition writes

$$\hat{g}^{f,i}_{NP}(\overline{x}_{\overline{\mathcal{B}}}) = |\overline{x}^i_{\overline{\mathcal{B}}} - \overline{c}_f| \geq 0, \ \overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}^i_{\overline{\mathcal{B}}}, \ i \in \{1,2,3\}.$$

For a better understanding, we consider the case of a flat contact surface parallel to the x-y-plane in more detail. Here, the contact surface is defined as

$$\overline{\Gamma}_{\overline{\mathcal{T}}} = \overline{\Gamma}^3_{\overline{\mathcal{T}}} = \left\{\overline{p} = \left[\overline{p}^1, \overline{p}^2, \overline{p}^3\right] \in \mathcal{T} \;\middle|\; \overline{p}^3 = \overline{c}_f =: \overline{h}\right\}$$

and the constant $\overline{c}_f$ represents the height $\overline{h}$ of the contact surface. Independently of the finite element node $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\mathcal{B}}$, the outward normal writes $\overline{n}^{\min}_f = [0,0,\pm 1]^T = \pm e_3$. We obtain $\overline{n}^{\min}_f = e_3$ if the deformable body is placed above the rigid surface and $\overline{n}^{\min}_f = -e_3$ for the opposite scenario. The related minimum distance point for a finite element node $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\mathcal{B}}$ writes $\overline{x}^{\min}_f = \left[\overline{x}^1_{\overline{\mathcal{B}}}, \overline{x}^2_{\overline{\mathcal{B}}}, \overline{h}\right]$ and we obtain the alternative non-penetration condition

$$\hat{g}^f_{NP}(\overline{x}_{\overline{\mathcal{B}}}) = \begin{cases} \overline{x}^3_{\overline{\mathcal{B}}} - \overline{h} \geq 0, & \text{rigid surface below deformable body} \\ \overline{x}^3_{\overline{\mathcal{B}}} - \overline{h} \leq 0, & \text{rigid surface above deformable body} \end{cases}, \ \overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}}.$$

Contact with the Curved Area of the Die Finally, we consider the curved part of the die as a rigid contact surface. This scenario is somehow similar to the case of contact with a rigid punch. Of course, we do not have to compute the distance to the center $\overline{c}_s$ of the rigid punch but to the center $\overline{c}_{\text{ref}}$ of a circle that is a specific slice of the curved edge of the die. The computation of $\overline{c}_{\text{ref}}$ has to be performed for each finite element node $\overline{x}_{\overline{\mathcal{B}}} = \left[\overline{x}^1_{\overline{\mathcal{B}}}, \overline{x}^2_{\overline{\mathcal{B}}}, \overline{x}^3_{\overline{\mathcal{B}}}\right] \in \overline{\Gamma}_{\overline{\mathcal{B}}}$ separately. It depends on the current position of the center $\overline{c}_s = \left[\overline{c}^1_s, \overline{c}^2_s, \overline{c}^3_s\right]$ of the rigid punch and the height $\overline{h}_d$ of the flat part of the die as well as on the radius $\overline{r}_d$, where the flat part of the die starts; see Figures 2.1 and 2.4. In our test setup we have $\overline{r}_d = 65$ mm. The final coordinates of $\overline{c}_{\text{ref}}$ write

$$\overline{c}_{\text{ref}} = \overline{c}_{\text{ref}}(\overline{x}_{\overline{\mathcal{B}}}) = \begin{pmatrix} \overline{c}^1_s + \frac{\overline{x}^1_{\overline{\mathcal{B}}} - \overline{c}^1_s}{\sqrt{(\overline{x}^1_{\overline{\mathcal{B}}} - \overline{c}^1_s)^2 + (\overline{x}^2_{\overline{\mathcal{B}}} - \overline{c}^2_s)^2}} \cdot \overline{r}_d \\ \overline{c}^2_s + \frac{\overline{x}^2_{\overline{\mathcal{B}}} - \overline{c}^2_s}{\sqrt{(\overline{x}^1_{\overline{\mathcal{B}}} - \overline{c}^1_s)^2 + (\overline{x}^2_{\overline{\mathcal{B}}} - \overline{c}^2_s)^2}} \cdot \overline{r}_d \\ \overline{r}_d + \overline{h}_d \end{pmatrix}.$$

Once the reference point $\overline{c}_{\text{ref}}$ is computed, we can proceed similar to the case with the rigid punch. Therefore, we additionally require the die radius $\overline{r}_e$ that is 10 mm in our test setup; see Figure 2.1. The outward normal has the same direction as the difference of the finite element node $\overline{x}_{\overline{\mathcal{B}}} \in \overline{\mathcal{B}}$ and the center of the slice, which is the reference point $\overline{c}_{\text{ref}}$. Thus, the outward normal writes $\overline{n}_c^{\min} = \frac{\overline{x}_{\overline{\mathcal{B}}} - \overline{c}_{\text{ref}}}{||\overline{x}_{\overline{\mathcal{B}}} - \overline{c}_{\text{ref}}||}$. The related minimum distance point is the orthogonal projection of the finite element node $\overline{x}_{\overline{\mathcal{B}}}$ onto the boundary of the slice, i.e., we obtain $\overline{x}_c^{\min} = \overline{c}_{\text{ref}} + \overline{r}_e \cdot \overline{n}_c^{\min}$. The alternative non-penetration condition for the curved part of the die writes

$$\hat{g}_N^c(\overline{x}_{\overline{\mathcal{B}}}) = ||\overline{x}_{\overline{\mathcal{B}}} - \overline{c}_{\text{ref}}|| - \overline{r}_e \geq 0, \ \overline{x}_{\overline{\mathcal{B}}} \in \overline{\Gamma}_{\overline{\mathcal{B}}},$$

where $\overline{c}_{\text{ref}}$ again depends on the finite element node $\overline{x}_{\overline{\mathcal{B}}}$.

5 Nonlinear FETI-DP and BDDC Methods

In this chapter, we focus on the introduction of a unified framework for nonlinear FETI-DP and BDDC methods. These methods can be used to solve discretized nonlinear partial differential equations. Different variants of nonlinear FETI-DP (Finite Element Tearing and Interconnecting - Dual-Primal) and BDDC (Balancing Domain Decomposition by Constraints) methods were introduced for the first time in [76, 77]. The unified framework was first published in [85], where this chapter is partially based on. We consider different examples of nonlinear FETI-DP methods (see Section 5.4) and compare them to the traditional Newton-Krylov-FETI-DP approach; see Section 5.5. Furthermore, problem dependent nonlinear FETI-DP methods are presented for the first time by exploiting information from the nonlinear residual, similarly to a strategy published in [50]; see Section 5.4.2.5. Additionally, we also shortly present a nonlinear BDDC approach (see Section 5.6.2) and discuss a strategy to make nonlinear domain decomposition methods more robust; cf. Section 5.7. Finally, we show numerical results for the introduced FETI-DP methods for different model problems such as the p-Laplace equation and hyperelasticity problems in two dimensions with and without contact. We show results obtained with sequential MATLAB [120] computations as well as parallel results; see Chapter 6.

5.1 Basic Notation

Before we start with the description of the different methods, let us first introduce the notation which is used throughout this chapter. It is similar to the standard notation used, e.g., in [89, 97, 98, 150]. We consider finite element methods which are based on the concept of divide and conquer, i.e., the computational domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is divided into N nonoverlapping subdomains Ω_i, $i = 1, \ldots, N$, where each subdomain Ω_i again is subdivided into finite elements. Note that we consider matching finite element nodes on the boundaries of neighboring subdomains. Denoting the closures of the com-

putational domain Ω and the subdomains Ω_i with $\overline{\Omega}$ and $\overline{\Omega}_i$, respectively, we obtain

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i.$$

The diameter of each subdomain Ω_i is denoted with H_i and $H := \max_{i=1,\dots,N} H_i$ is the maximum diameter of all subdomains. Similar, the maximum diameter of the finite elements is denoted with h.

Usually, on some parts of the boundary of the computational domain, the solution is prescribed by some Dirichlet boundary conditions. We refer to this part of the boundary $\partial\Omega$ as $\partial\Omega_D$. The domain decomposition interface Γ consists of all points, which belong to the closure of at least two different subdomains but do not belong to the Dirichlet boundary, i.e., $\Gamma = \left\{ x \in (\overline{\Omega}_i \cap \overline{\Omega}_j) \setminus \partial\Omega_D \;\middle|\; i \neq j \right\}$; see also [106].

In the following, we consider the discretized version of the computational domain Ω, which is referred to as Ω^h. Since we consider matching finite element nodes on neighboring subdomains, all physical points $x \in \Gamma$ belong to the finite element discretization of at least two different subdomains. Analogously to [106], we denote with Γ^h, $\partial\Omega_D^h$, and $\partial\Omega_i^h$, $i = 1, \dots, N$, the sets of finite element nodes belonging to Γ, $\partial\Omega_D$, and $\partial\Omega_i$, respectively. For simplicity, the superscript h is neglected in the following.

To follow the standard FETI-DP notation as in, e.g., [89, 97, 98, 150], we denote the local finite element spaces by W_i and the product space by $W = W_1 \times \cdots \times W_N$. Furthermore, we denote the set of finite element functions which are continuous in all interface variables by $\widehat{W} \subset W$. The space of globally assembled finite element functions is denoted by $V^h = V^h(\Omega)$; it is isomorphic to $\widehat{W}$. Restrictions from the globally assembled finite element functions V^h to the local subdomains are performed by the operator

$$R_i : V^h \to W_i, \; i = 1, \dots, N.$$

All methods that are considered throughout this thesis make use of a strong coupling in some degrees of freedom. Therefore, we partition the degrees of freedom into different subsets. The interface variables Γ^h are divided into dual (Δ) and primal (Π) variables. Additionally, we introduce the set of interior variables (I), which contains all variables that do not belong to Γ^h. Combining the inner and dual interface variables, we obtain $B = [I, \Delta]$. Similar to $\widehat{W}$, which contains all finite element functions that are continuous across the interface, we introduce the space $\widetilde{W}$, which contains all finite element functions that are

continuous only in the primal variables. Since the primal variables are a subset of the interface variables, we have

$$\widehat{W} \subset \widetilde{W} \subset W.$$

Throughout this thesis, we usually consider a nonlinear system of equations, which results from the finite element discretization of an energy functional. For example, we consider a nonlinear problem of the form $K(u) - f = 0$. In this case, the tangential matrix $DK(u)$ occurs in the linearized system. In order to use a consistent notation throughout this thesis, also the system matrix of a linear problem is denoted with DK. This is in contrast to the standard notation but identical to the notation in [107].

5.2 Classical FETI-DP

We recall the traditional FETI-DP approach for a better readability of the following sections. The content of this section is strongly based on the descriptions in, e.g., [107, Sec. 2.1] and [94].

As mentioned before, the FETI-DP method can be used to solve linear or linearized systems of equations which result from the finite element discretization of a partial differential equation. It was first published in [38]. In general, the solution is obtained by solving the equivalent minimization problem of the corresponding energy functional.

If the resulting linearized system of equations on the computational domain Ω leads to a large problem

$$D\widehat{K}\ \hat{u} = \hat{f},\ \hat{u} \in V^h,$$

which cannot be solved in a direct fashion, we have to apply other solution strategies. The idea is to decompose the domain Ω into N nonoverlapping subdomains of a sufficient size. Subsequently, on each subdomain Ω_i we solve a local problem

$$DK^{(i)}\ u^{(i)} = f^{(i)} \tag{5.1}$$

and the overall solution is derived by interconnection of the local solutions on the interface. The local stiffness matrices $DK^{(i)}$ and the local right-hand sides $f^{(i)}$ result from local finite element assemblies on the subdomains. Note that the subdomain problems can be solved in parallel.

Since finite element nodes in Γ^h have at least one local copy in another subdomain, we have to ensure continuity on the interface. In FETI-DP methods, we use a combination of global subassembly in the primal variables Π (e.g. subdomain vertices) and additional dual conditions on the remaining dual variables Δ; see Figure 5.3.

For the continuity in the dual interface variables, we introduce the FETI-DP jump operator B_B (see, e.g., [94, 150]) and additional Lagrange multipliers λ to enforce the zero jump condition $B_B u_B = 0$, which is also called continuity condition. Note that B_B is not unique and its number of rows depends on the number of Lagrange multipliers. For example, for a subdomain corner belonging to four subdomains in 2D, three up to six Lagrange multipliers can be implemented without changing the solution. Likewise, for an edge in three dimensions, the choice of Lagrange multipliers is not unique ; see [150]. The usage of a minimal set of Lagrange multipliers is referred to as the nonredundant case and using all possible Lagrange multipliers is called the fully redundant case, which is often used in practice; see, e.g., [106,107]. Therefore, we have one Lagrange multiplier λ for each pair of local copies of a global interface variable. Since we have continuity in the inner variables, i.e., the jump is zero, the jump operator B_B writes $B_B = \begin{bmatrix} 0_I & B_\Delta \end{bmatrix}$. Each row of B_Δ belongs to one Lagrange multiplier, i.e., each row of $B_B u_B = 0$ enforces equality of two local copies of a global interface variable. Therefore, each row in B_Δ only contains a single 1 and a single -1 and a multiplication with B_B only causes nearest neighbor communication.

For the global assembly in the primal variables Π, we introduce the operator $R_\Pi^T = \begin{bmatrix} R_\Pi^{(1)^T}, \dots, R_\Pi^{(N)^T} \end{bmatrix}$, which is standard in FETI-DP; see, e.g., [94,150]. For example, a typical choice for the primal variables are the subdomain vertices. Let us note that further choices of primal variables are possible, such as edge constraints; see [97]. Furthermore, we can also choose a problem-dependent set of primal variables by using an adaptive coarse space; see, e.g., [58,76,106]. Subassembly in the primal variables leads to the FETI-DP coarse problem which is a global problem and generally all subdomains add primal constraints. In the following, all quantities which are subassembled in the primal variables are marked with a tilde. In comparison to the discretized problem on the whole domain Ω, the FETI-DP coarse problem is significantly smaller. Furthermore, computations on the FETI-DP coarse problem can be parallelized using inexact FETI-DP variants; see Section 5.4.4 and [107, Sec. 3].

Let us go back to Equation (5.1). Using the introduced partition of variables in Section 5.1, the local subdomain matrices $DK^{(i)}$ write (after reordering)

$$DK^{(i)} := \begin{bmatrix} DK_{BB}^{(i)} & DK_{B\Pi}^{(i)} \\ DK_{\Pi B}^{(i)} & DK_{\Pi\Pi}^{(i)} \end{bmatrix}, \tag{5.2}$$

where the upper left block $DK_{BB}^{(i)}$ writes

$$DK_{BB}^{(i)} := \begin{bmatrix} DK_{II}^{(i)} & DK_{I\Delta}^{(i)} \\ DK_{\Delta I}^{(i)} & DK_{\Delta\Delta}^{(i)} \end{bmatrix}. \tag{5.3}$$

Note that $DK_{\Pi B}^{(i)} = DK_{\Pi B}^{(i)^T}$ and $DK_{\Delta I}^{(i)} = DK_{\Delta I}^{(i)^T}$ hold.

Subassembly in the primal variables leads to the FETI-DP master system

$$\begin{bmatrix} DK_{BB} & D\widetilde{K}_{B\Pi} & B_B^T \\ D\widetilde{K}_{\Pi B} & D\widetilde{K}_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}, \tag{5.4}$$

where we have used

$$\begin{aligned} DK_{BB} &:= \begin{bmatrix} DK_{BB}^{(1)} & & \\ & \ddots & \\ & & DK_{BB}^{(N)} \end{bmatrix}, \\ D\widetilde{K}_{\Pi B} &:= \left[D\widetilde{K}_{\Pi B}^{(1)}, \ldots, D\widetilde{K}_{\Pi B}^{(N)} \right] = \left[R_\Pi^{(1)^T} DK_{\Pi B}^{(1)}, \ldots, R_\Pi^{(N)^T} DK_{\Pi B}^{(N)} \right], \text{ and} \\ D\widetilde{K}_{\Pi\Pi} &:= \sum_{i=1}^{N} R_\Pi^{(i)^T} DK_{\Pi\Pi}^{(i)} R_\Pi^{(i)}. \end{aligned} \tag{5.5}$$

Note that the diagonal blocks $DK_{BB}^{(i)}$, $i = 1, \ldots, N$, are completely local to the subdomains since they are restricted to the interior and dual variables of the local matrices $DK^{(i)}$. Obviously, the matrix B_B as well as the vectors u_B and f_B have an analogous block structure, i.e.,

$$B_B = \left[B_B^{(1)}, \ldots, B_B^{(N)} \right], \; u_B = \left[u_B^{(1)^T}, \ldots, u_B^{(N)^T} \right]^T, \; f_B = \left[f_B^{(1)^T}, \ldots, f_B^{(N)^T} \right]^T.$$

Defining the Schur complement $\widetilde{S}_{\Pi\Pi}$ in the primal variables as

$$\widetilde{S}_{\Pi\Pi} := D\widetilde{K}_{\Pi\Pi} - D\widetilde{K}_{\Pi B} DK_{BB}^{-1} D\widetilde{K}_{B\Pi} \tag{5.6}$$

and using the notation

$$F = B_B DK_{BB}^{-1} B_B^T + B_B DK_{BB}^{-1} D\widetilde{K}_{B\Pi} \widetilde{S}_{\Pi\Pi}^{-1} \widetilde{K}_{\Pi B} DK_{BB}^{-1} B_B^T \quad \text{and}$$
$$d = B_B DK_{BB}^{-1} f_B + B_B K_{BB}^{-1} D\widetilde{K}_{B\Pi} \widetilde{S}_{\Pi\Pi}^{-1} \left(\tilde{f}_\Pi - \widetilde{K}_{\Pi B} DK_{BB}^{-1} f_B \right),$$

we finally obtain

$$F\lambda = d \tag{5.7}$$

from the third line of system (5.4) by block Gauß elimination of the variables u_B and $\tilde{u}_\Pi$. Note that inverting DK_{BB} - due to its block structure - only requires inverting the local operators $DK_{BB}^{(i)}$; see Equation (5.5).

If we combine the variables u_B and $\tilde{u}_\Pi$ to $\tilde{u} := \left[u_B^T, \tilde{u}_\Pi^T\right]^T$ as well as f_B and $\tilde{f}_\Pi$ to $\tilde{f} := \left[f_B^T, \tilde{f}_\Pi^T\right]^T$ and write $B := \left[B_B \ 0\right]$, we can rewrite Equation (5.4) in compressed form as

$$\begin{bmatrix} D\widetilde{K} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ 0 \end{bmatrix}. \tag{5.8}$$

Here, $D\widetilde{K}$ is defined as

$$D\widetilde{K} := \begin{bmatrix} DK_{BB} & D\widetilde{K}_{B\Pi} \\ D\widetilde{K}_{\Pi B} & D\widetilde{K}_{\Pi\Pi} \end{bmatrix}. \tag{5.9}$$

The equivalent version of Equation (5.7) in the compressed form (cf. Equation (5.8)) writes

$$B(D\widetilde{K})^{-1} B^T \lambda = B(D\widetilde{K})^{-1} f. \tag{5.10}$$

Using the assembly operator R_Π^T, we can also write

$$D\widetilde{K} = R_\Pi^T (DK) R_\Pi,$$

where DK has a block-diagonal structure with $DK^{(i)}$, $i = 1, \ldots, N$, (cf. Equation (5.2)) on its diagonal block entries.

Instead of solving $F\lambda = d$ (cf. Equation (5.7)) or its equivalent compressed version (cf. Equation (5.10)), the FETI-DP method solves the preconditioned system

$$M^{-1} F\lambda = M^{-1} d \tag{5.11}$$

iteratively using a Krylov subspace method, such as CG [60] or GMRES [51, 130]. There are different possible choices of the preconditioner M^{-1} such as the standard Dirichlet preconditioner (see [41, 150])

$$M^{-1}_{\text{FETI}_D} := \sum_{i=1}^{N} B^{(i)}_{\Delta,D} S^{(i)}_{\Delta\Delta} B^{(i)^T}_{\Delta,D}, \tag{5.12}$$

which is a weighted sum of local Schur complements

$$S^{(i)}_{\Delta\Delta} := DK^{(i)}_{\Delta\Delta} - DK^{(i)}_{\Delta I}(DK^{(i)}_{II})^{-1} DK^{(i)}_{I\Delta},$$

which result from the elimination of the interior variables I in $DK^{(i)}_{BB}$; see Equation (5.3). Replacing the local Schur complements in Equation (5.12) by local stiffness matrices on the dual interface leads to the lumped preconditioner, which is defined as

$$M^{-1}_{\text{FETI}_L} := \sum_{i=1}^{N} B^{(i)}_{\Delta,D} K^{(i)}_{\Delta\Delta} B^{(i)^T}_{\Delta,D}; \tag{5.13}$$

see, e.g., [94, 150]. The operators $B^{(i)}_{\Delta,D}$, $i = 1, \ldots, N$, result from the jump operators $B^{(i)}_B$ by restriction to the dual variables Δ and scaling of the rows. For more details including different choices of weights, we refer to [94, 150]. By construction, the considered preconditioners are weighted sums of local operators. Thus, they can be applied completely in parallel. Throughout this thesis, we only use the Dirichlet preconditioner, i.e., $M^{-1} := M^{-1}_{\text{FETI}_D}$, for which the polylogarithmic condition number bound

$$\kappa\left(M^{-1}F\right) \leq C\left(1 + \log\left(\frac{H}{h}\right)\right)^2 \tag{5.14}$$

has been shown for various two- and three-dimensional model problems; see [90, 95, 97, 98, 118]. Here, the constant C is independent of the jumps in the coefficients of the PDE as well as of the parameters H and h. Note that the condition number bound for the lumped preconditioner contains a linear factor $\frac{H}{h}$ instead of the polylogarithmic bound; see [94] and the references therein.

For each problem, a sufficient set of primal variables has to be chosen and suitable weights for the preconditioner have to be found depending on the coefficients of the partial differential equation.

Let us further note that we mostly use the compressed version of the FETI-DP master system (see Equation (5.8)) instead of Equation (5.4) in the remainder of this thesis.

5.3 Nonlinear Domain Decomposition in the Context of FETI-DP

The FETI-DP approach, as introduced above, can only be applied to a linear system of equations, but most simulations of real world problems require the solution of nonlinear systems of equations. To tackle nonlinear problems with the FETI-DP approach, the nonlinear system can be either linearized before decomposition or we can perform a nonlinear decomposition before linearization. Throughout this thesis, we exclusively use Newton's method for the linearization of nonlinear problems due to its fast convergence in a neighborhood of the solution. Applying Newton's method to the nonlinear problem before decomposition leads to the Newton-Krylov-FETI-DP approach; see Section 5.5. Otherwise, decomposing the nonlinear problem and applying Newton's method to the local nonlinear problems results in nonlinear FETI-DP methods, which are introduced in Section 5.4.

Before we introduce a unified framework of nonlinear FETI-DP methods, we first recall the general ideas and assumptions of nonlinear domain decomposition. The discussion in this section is strongly based on the descriptions in [107, Sec. 2.3, Sec. 2.5.1] and [77, Sec. 2, Sec. 3.1].

As already mentioned in the previous sections, we do not consider the discrete problem of the partial differential equation, but we solve the equivalent minimization problem

$$\min_{\hat{u}\in V^h} J(\hat{u}) \tag{5.15}$$

of the corresponding energy functional $J : V^h \to \mathbb{R}$, which is usually nonlinear. This leads to the (nonlinear) system of equations

$$A(\hat{u}) = 0, \quad \hat{u} \in V^h. \tag{5.16}$$

We assume that the global energy functional J can be represented as a sum of local energy functionals, which operate on the subdomains. This assumption is generally fulfilled for standard problems discretized by finite elements due to the additivity of the integral and it is summarized in Assumption 1; see also [77, 107].

Assumption 1. *There exist local energy functionals $J_i : W_i \to \mathbb{R}$, $i = 1, \dots, N$, such that the global energy functional $J(\hat{u})$ can be represented as a sum of the*

local energies, i.e.,

$$J(\hat{u}) = \sum_{i=1}^{N} J_i(u_i) \quad \forall \hat{u} \in V^h.$$

Here, $u_i := R_i \hat{u}$ are the restrictions of $\hat{u}$ to the subdomains.

This assumption holds for the p-Laplace problem as well as for standard hyperelasticity problems that are considered in the remainder of this thesis. It is also fulfilled for incompressible hyperelasticity problems and other relevant nonlinear problems, but might not be satisfied for problems with nonlocal phenomena, e.g., nonlocal damage models in structural mechanics; see [77].

As before, we introduce a linear, discrete jump operator $\overline{B} = \left[\overline{B}_1, \ldots, \overline{B}_N\right]$ to enforce continuity across the interface Γ; see Section 5.2. So far, we do not consider a strong coupling in the primal variables Π. Therefore, the jump operator $\overline{B}$ also enforces continuity across primal variables (Π) and not only on the dual variables (Δ). Consequently, the jump operator in Section 5.2 can be derived from $\overline{B}$ by simply removing all lines corresponding to Π; see [107].

For any $u = \left[u_1^T, \ldots, u_N^T\right]^T \in W$, $u_i \in W_i$, with $\overline{B}u = 0$, we have $u \in \widehat{W}$. Recalling that $\widehat{W}$ and V^h are isomorphic, we can rewrite the minimization problem (5.15) with Assumption 1 and obtain

$$\min_{\hat{u} \in V^h} J(\hat{u}) = \min_{\substack{\hat{u} \in V^h \\ u_i = R_i \hat{u}}} \sum_{i=1}^{N} J_i(u_i) = \min_{u \in \widehat{W}} \sum_{i=1}^{N} J_i(u_i) = \min_{\substack{u \in W \\ \overline{B}u = 0}} \sum_{i=1}^{N} J_i(u_i); \tag{5.17}$$

see also [107, Sec. 2].

We introduce the space of Lagrange multipliers as $V := \operatorname{range}(\overline{B})$ and the Lagrange function

$$\begin{aligned} \mathcal{L} &: \; W \times V \to \mathbb{R}, \\ \mathcal{L}(u, \lambda) &= \sum_{i=1}^{N} J_i(u_i) + (\overline{B}u)^T \lambda. \end{aligned} \tag{5.18}$$

The computation of the stationary points of Equation (5.18) results in taking the partial derivatives of the Lagrange functional with respect to u and λ and setting the resulting equations equal to zero. The stationary points of Equation (5.18) are then the solutions of the system

$$\begin{bmatrix} \sum_{i=1}^{N} J_i'(u_i)(v_i) + (\overline{B}v)^T \lambda \\ (\overline{B}u)^T \mu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \forall v \in W, \; \mu \in V. \tag{5.19}$$

We assume that we have N_i nodal finite element basis functions for each local finite element space W_i, $i = 1, \dots, N$, which are denoted by $\varphi_{i,j}$, $j = 1, \dots, N_i$. Furthermore, we assume that we can represent the evaluations of the derivative of the local energy functionals in the nodal basis functions $J_i'(u_i)(\varphi_{i,j})$ by $(K_i(u_i) - f_i)_j$. Using the notation

$$K(u) := \begin{bmatrix} K_1(u_1) \\ \vdots \\ K_N(u_N) \end{bmatrix}, \qquad f := \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}, \text{ and } \qquad u := \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}, \tag{5.20}$$

we derive the discrete nonlinear system of equations

$$\begin{aligned} K(u) + \overline{B}^T \lambda &= f, \\ \overline{B} u &= 0 \end{aligned} \tag{5.21}$$

from Equation (5.19). Note that f_i is independent of u_i and that Equation (5.21) can be seen as the nonlinear analogon of the linear FETI-DP master system in Equation (5.8). The equivalence of system (5.21) and the initial problem (5.15) is further discussed in [107, Sec. 2.5.1] and [77, Sec. 3.1].

Let us now have a closer look at the nonlinear system (5.21). As before, we partition the interface variables u_Γ into a set of primal variables $\tilde{u}_\Pi$ and u_Δ; see also Section 5.2. To consider the partially assembled nonlinear system of equations, we recall the definitions of the jump operator B from Section 5.2, which enforces continuity in the dual variables u_Δ, and of the partial assembly operator R_Π^T. Again, note that the operator B from Section 5.2 can be derived from $\overline{B}$ by removing all lines corresponding to the primal variables. Our nonlinear FETI-DP methods are constructed by partial assembly of $K_i(u_i)$ and f_i in the primal variables. They are based on the nonlinear FETI-DP master system

$$\begin{aligned} R_\Pi^T K(R_\Pi \tilde{u}) + B^T \lambda - R_\Pi^T f &= 0, \\ B \tilde{u} &= 0. \end{aligned} \tag{5.22}$$

Introducing the notation

$$\widetilde{K}(\tilde{u}) := R_\Pi^T K(R_\Pi \tilde{u}) \quad \text{ and } \quad \tilde{f} := R_\Pi^T f,$$

Equation (5.22) can be written in more compact form as

$$A(\tilde{u}, \lambda) := \begin{bmatrix} \widetilde{K}(\tilde{u}) + B^T \lambda - \tilde{f} \\ B \tilde{u} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5.23}$$

The computation of the derivative of $\widetilde{K}(\tilde{u})$ yields with the chain rule

$$D\widetilde{K}(\tilde{u}) := D(\widetilde{K}(\tilde{u})) = D(R_\Pi^T K(R_\Pi \tilde{u})) = R_\Pi^T DK(R_\Pi \tilde{u}) R_\Pi. \tag{5.24}$$

From Equation (5.24), we obtain that the derivative of the subassembled nonlinear operator $\widetilde{K}$ can be computed by partially assembling DK, i.e., by partially assembling the derivatives of the local subdomain operators K_i, $, i = 1, \ldots, N$.

Instead of simply linearizing system (5.23), we first apply a nonlinear right-preconditioner $M(\tilde{u}, \lambda)$ which, in case it is the identity, is called Nonlinear-FETI-DP-1; see Section 5.4.2.1. The application of the preconditioner $M(\tilde{u}, \lambda)$ is associated with a nonlinear elimination process [108] and different choices of M lead to different variants of nonlinear FETI-DP. Furthermore, our preconditioners M can also be viewed in the context of nonlinear right-preconditioning [20, 23] in Newton's method.

After the application of the preconditioner $M(\tilde{u}, \lambda)$ to Equation (5.23), the resulting system

$$A(M(\tilde{u}, \lambda)) = 0 \tag{5.25}$$

is linearized using Newton's method. As in the classical NK-FETI-DP approach (see Section 5.5), in each Newton iteration the linearized system is solved with a Krylov subspace method, such as CG [60] or GMRES [51, 130], using a suitable preconditioner, such as the Dirichlet preconditioner; see [41, 150] and also Section 5.2 for its definition.

Applying Newton's method to Equation (5.25) yields the iteration

$$\begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} := \begin{bmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix}, \tag{5.26}$$

where $\alpha^{(k)}$ is a suitable step length and $\delta\tilde{u}^{(k)}$ and $\delta\lambda^{(k)}$ are the Newton updates. Both are computed from

$$\Big(DA(M(\tilde{u}^{(k)}, \lambda^{(k)})) \cdot DM(\tilde{u}^{(k)}, \lambda^{(k)}) \Big) \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = A(M(\tilde{u}^{(k)}, \lambda^{(k)}), \tag{5.27}$$

where the left-hand side is the derivative of Equation (5.25) with respect to $\tilde{u}$ and λ. Note that the superscript $\cdot^{(k)}$ no longer indicates the corresponding subdomain Ω_k, as is the case in Section 5.2, but the current Newton iteration. Instead, the associated subdomain Ω_j is marked with an index j in the following.

	Nonlinear FETI-DP	Nonlinear BDDC
1.	Mapping $M : \widetilde{W} \times V \to \widetilde{W} \times V$	Mapping $M : \widehat{W} \to \widehat{W}$
2.	M puts the current iterate into the neighborhood of the solution; see also [18].	
3.	$M(x)$ is easily computable compared to the inverse action of $A(x)$ with $x := (\tilde{u}, \lambda)$.	$M(x)$ is easily computable compared to the inverse action of $A(x)$ with $x := \hat{u}$.

Figure 5.1: Properties of the nonlinear preconditioner M; see also [85, Fig. 1 and Fig. 4]. **Left:** Nonlinear FETI-DP methods. **Right:** Nonlinear BDDC methods.

In each Newton iteration, the nonlinear preconditioner is evaluated by computing

$$g^{(k)} := M(\tilde{u}^{(k)}, \lambda^{(k)}). \tag{5.28}$$

Let us note that we are in fact not interested in obtaining $\tilde{u}^{**}$ and λ^{**} satisfying $A(M(\tilde{u}^{**}, \lambda^{**})) = 0$, but, as usual in nonlinear right-preconditioning, we are rather interested in the solution $(\tilde{u}^*, \lambda^*) = M(\tilde{u}^{**}, \lambda^{**})$.

To ensure that the application of the nonlinear preconditioner M accelerates the computation of the solution, we list some assumptions and desired properties in Figure 5.1 (left).

Let us give some remarks on the convergence of nonlinear FETI-DP methods. First, we consider the convergence of Newton's method. Therefore, we assume that problem (5.16) has a solution $(\tilde{u}^*, \lambda^*)$, i.e., $A(\tilde{u}^*, \lambda^*) = 0$ holds. We formulate the following assumption; see also [77, 85, 107].

Assumption 2. *Let U be an open neighborhood of the solution $(\tilde{u}^*, \lambda^*)$. We assume that $A(\tilde{u}, \lambda)$ is continuously differentiable in U and that $DA(\tilde{u}^*, \lambda^*)$ is a nonsingular matrix.*

If Assumption 2 is satisfied, Newton's method converges to the solution $(\tilde{u}^*, \lambda^*)$ of $A(\tilde{u}, \lambda) = 0$ for all $(\tilde{u}^{(0)}, \lambda^{(0)}) \in U^* \subset U$; see, e.g., [125, Sec 10.2.2]. Note that the subset U^* is obtained from the inverse function theorem (see [46, Sec. 18, Th. 3, p. 100]). If $DA(\tilde{u}^*, \lambda^*)$ is nonsingular, as by Assumption 2, $D\widetilde{K}$ is also nonsingular in a neighborhood of $\tilde{u}^*$ if a suitable set of primal constraints is used. Note that $DA(\tilde{u}^*, \lambda^*)$ will be singular in the case of redundant Lagrange multipliers since the matrix B only has full rank for nonredundant multipliers. Nonetheless, the same Newton iterates $\tilde{u}^{(k)}$ are

derived when using redundant Lagrange multipliers, which yields the same convergence behavior; see [107, Sec. 2.5.2].

As mentioned before, redundant Lagrange multipliers are often used in practice. For this case, stopping criteria based on the update $\delta\lambda^{(k)}$ have to be avoided since convergence can only be guaranteed in $\widetilde{W}$; see the end of [107, Sec. 2.5.2] as well as [85, Sec. 2.3].

In order to analyze the convergence behavior of nonlinear FETI-DP methods for solving the nonlinear right-preconditioned system (5.25), we make the following additional assumption; see also [85, Ass. 2.2].

Assumption 3. *Let V^* be an open neighborhood of the solution $(\tilde{u}, \lambda)$. The evaluation of the nonlinear right-preconditioner $M(\tilde{u}, \lambda)$ is well defined and computable in V^* and $M(V^*) \subset U^*$.*

Assumption 3 yields that $M(\tilde{u}^{(k)}, \lambda^{(k)}) \in U^*$ for all iterates $(\tilde{u}^{(k)}, \lambda^{(k)}) \in V^*$. It follows with Assumption 2 and the discussion above that the nonlinear FETI-DP method for the nonlinear right-preconditioned system (5.25) converges to the solution $(\tilde{u}^*, \lambda^*)$ for all initial values $(\tilde{u}^{(0)}, \lambda^{(0)}) \in V^*$. If V^* is larger than U^*, the convergence radius is increased.

5.4 Unified Framework for Nonlinear FETI-DP

In this section, we now present a common framework for different nonlinear FETI-DP methods [76, 77, 84] using partial nonlinear elimination. Our representations are based on those in [85], where the framework was first presented.

Similar to classical FETI-DP methods, where an index splitting is used for the partial assembly, this splitting can also be used for the construction of the nonlinear preconditioner M. As mentioned before, the effect of the nonlinear preconditioner can be understood as a nonlinear partial elimination process. Therefore, we introduce the variable splitting

$$\tilde{u} = \begin{bmatrix} \tilde{u}_{\mathcal{E}} & \tilde{u}_{\mathcal{L}} \end{bmatrix},$$

where the index $\mathcal{E}$ denotes the set of variables which will be eliminated nonlinearly and $\mathcal{L}$ is the set of variables which will be linearized. Usually, the elimination set is denoted with E instead of $\mathcal{E}$ (see [85, 86]), but we have chosen the notation $\mathcal{E}$ to distinguish it from Young's modulus E in elasticity; see Section 6.1.2. To use a consistent notation, we also use $\mathcal{L}$ instead of L for the set of variables that are not eliminated nonlinearly.

Analogously to $\tilde{u}$, the jump operator B can be written in terms of the index sets $\mathcal{E}$ and $\mathcal{L}$ and we obtain

$$B = \begin{bmatrix} B_{\mathcal{E}} & B_{\mathcal{L}} \end{bmatrix}.$$

Finally, the unpreconditioned system (5.23) writes

$$A(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) := \begin{bmatrix} A_{\mathcal{E}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) \\ A_{\mathcal{L}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) \\ B_{\mathcal{E}}\tilde{u}_{\mathcal{E}} + B_{\mathcal{L}}\tilde{u}_{\mathcal{L}} \end{bmatrix} = \begin{bmatrix} \widetilde{K}_{\mathcal{E}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}) + B_{\mathcal{E}}^T\lambda - \tilde{f}_{\mathcal{E}} \\ \widetilde{K}_{\mathcal{L}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}) + B_{\mathcal{L}}^T\lambda - \tilde{f}_{\mathcal{L}} \\ B_{\mathcal{E}}\tilde{u}_{\mathcal{E}} + B_{\mathcal{L}}\tilde{u}_{\mathcal{L}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.29}$$

Since $\mathcal{E}$ is chosen such that it contains all variables which will be eliminated nonlinearly, the elimination process is restricted to the variables $\tilde{u}_{\mathcal{E}}$. Therefore, the nonlinear preconditioner $M(\tilde{u}, \lambda) = M(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda)$ is linear in $\tilde{u}_{\mathcal{L}}$ and λ. Here, in fact, the preconditioner is only the identity in $\tilde{u}_{\mathcal{L}}$ and λ. We introduce the notation

$$\begin{aligned} M(\tilde{u}, \lambda) = M(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) &:= (M_{\tilde{u}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda), \lambda) \\ &:= (M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda), \tilde{u}_{\mathcal{L}}, \lambda) \\ &= (M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda), \tilde{u}_{\mathcal{L}}, \lambda), \end{aligned} \tag{5.30}$$

Note that $M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) = (M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda), \tilde{u}_{\mathcal{L}}, \lambda)$ is independent of its first argument $\tilde{u}_{\mathcal{E}}$, which is only introduced for convenience, such that the derivative $DM_{\tilde{u}_{\mathcal{E}}}$ is a square matrix; see Section 5.4.1. Since we are interested in the nonlinear elimination of all variables in the index set $\mathcal{E}$, we consider the first line of Equation (5.29). All variables in $\mathcal{E}$ are eliminated if the right-preconditioner $M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda)$ fulfills the equation

$$\widetilde{K}_{\mathcal{E}}(M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda), \tilde{u}_{\mathcal{L}}) + B_{\mathcal{E}}^T\lambda - \tilde{f}_{\mathcal{E}} = 0, \tag{5.31}$$

i.e., $M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda)$ is defined implicitly from the first line of Equation (5.29); cf. [84, Eq. (5)]. Therefore, we have to solve Equation (5.31) for $M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda)$, e.g., with Newton's method, and subsequently replace the variables $\tilde{u}_{\mathcal{E}}$ in the second and third line of Equation (5.29) by $M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda)$. Let us note that the solution of Equation (5.31) is discussed in more detail in Section 5.4.1.

We obtain the nonlinear Schur complement system

$$S_{\mathcal{L}}(\tilde{u}_{\mathcal{L}}, \lambda) = 0, \tag{5.32}$$

where the nonlinear Schur complement $S_{\mathcal{L}}(\tilde{u}_{\mathcal{L}}, \lambda)$ is defined as

$$S_{\mathcal{L}}(\tilde{u}_{\mathcal{L}}, \lambda) := \begin{bmatrix} \widetilde{K}_{\mathcal{L}}(M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda), \tilde{u}_{\mathcal{L}}) + B_{\mathcal{L}}^T \lambda - \tilde{f}_{\mathcal{L}} \\ B_{\mathcal{E}} M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda) + B_{\mathcal{L}} \tilde{u}_{\mathcal{L}} \end{bmatrix}. \tag{5.33}$$

Accordingly, the nonlinearly preconditioned system (5.25) writes

$$A(M(\tilde{u}, \lambda)) = A(M(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda)) = A(M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}}, \lambda), \tilde{u}_{\mathcal{L}}, \lambda) = \begin{bmatrix} 0 \\ S_{\mathcal{L}}(\tilde{u}_{\mathcal{L}}, \lambda) \end{bmatrix}.$$

Finally, the nonlinear system (5.32) has to be solved, e.g. with Newton's method. The linearization of Equation (5.32) requires the computation of the tangent $DS_{\mathcal{L}}$ of $S_{\mathcal{L}}$, which is obtained using the chain rule and the implicit function theorem; see Section 5.4.1.

5.4.1 Computing the Tangent

Each application of the preconditioner M requires the solution of a nonlinear system

$$A_{\mathcal{E}}(g_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) = K_{\mathcal{E}}(g_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}) + B_{\mathcal{E}}^T \lambda - \tilde{f}_{\mathcal{E}} = 0 \tag{5.34}$$

for $g_{\mathcal{E}}$; cf. Equation (5.31). The computation of

$$g_{\mathcal{E}} := M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda) \tag{5.35}$$

can be achieved by applying Newton's method to Equation (5.34). This yields the iteration

$$g_{\mathcal{E},l+1}^{(k)} := g_{\mathcal{E},l}^{(k)} - \left(D_{\mathcal{E}} \widetilde{K}_{\mathcal{E}}(g_{\mathcal{E},l}^{(k)}, \tilde{u}_{\mathcal{L}}^{(k)}) \right)^{-1} \left(\widetilde{K}_{\mathcal{E}}(g_{\mathcal{E},l}^{(k)}, \tilde{u}_{\mathcal{L}}^{(k)}) + B_{\mathcal{E}}^T \lambda - \tilde{f}_{\mathcal{E}} \right), \tag{5.36}$$

which converges to the solution $g_{\mathcal{E}}^{(k)}$ under sufficient assumptions which are made throughout this thesis; see Section 5.3. Note that the indices l and k in Equation (5.36) represent the inner and outer Newton iteration, respectively. Similar to Equation (5.9), we assume the partition

$$D\widetilde{K}(\cdot) = \begin{bmatrix} D\widetilde{K}_{\mathcal{E}\mathcal{E}}(\cdot) & D\widetilde{K}_{\mathcal{E}\mathcal{L}}(\cdot) \\ D\widetilde{K}_{\mathcal{L}\mathcal{E}}(\cdot) & D\widetilde{K}_{\mathcal{L}\mathcal{L}}(\cdot) \end{bmatrix} \tag{5.37}$$

for the tangent of $\widetilde{K}$. Using the notation $g^{(k)} := M_{\tilde{u}}(\tilde{u}_{\mathcal{E}}^{(k)}, \tilde{u}_{\mathcal{L}}^{(k)}, \lambda^{(k)})$ we obtain

$$g^{(k)} = (g_{\mathcal{E}}^{(k)}, \tilde{u}_{\mathcal{L}}^{(k)}) \tag{5.38}$$

from Equation (5.30).

For the computation of the Newton update (cf. Equation (5.27)), we require the derivative of the preconditioner M. Since M is the identity in $\tilde{u}_\mathcal{L}$ and λ, we only have to consider the partial derivatives $M_{\tilde{u}_\mathcal{E}}$ with respect to the different variables $\tilde{u}_\mathcal{E}$, $\tilde{u}_\mathcal{L}$, and λ in more detail. Note that the partial derivatives of $M_{\tilde{u}_\mathcal{E}}$ are obtained from Equation (5.34) using the chain rule. The derivative of Equation (5.34) with respect to the first variable $\tilde{u}_\mathcal{E}$ yields

$$D_{\tilde{u}_\mathcal{E}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L})\cdot D_{\tilde{u}_\mathcal{E}}M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)=0. \tag{5.39}$$

Assuming that $D_{\tilde{u}_\mathcal{E}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L})$ is invertible (see Section 5.3), we obtain

$$D_{\tilde{u}_\mathcal{E}}M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)=0. \tag{5.40}$$

Computation of the derivative of Equation (5.34) with respect to $\tilde{u}_\mathcal{L}$ yields

$$\begin{aligned}&D_{\tilde{u}_\mathcal{L}}M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)\\&=-\left(D_{\tilde{u}_\mathcal{E}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L})\right)^{-1}\cdot D_{\tilde{u}_\mathcal{L}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L}).\end{aligned} \tag{5.41}$$

Again, invertibility of $D_{\tilde{u}_\mathcal{E}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L})$ is assumed.

The computation of the derivative of Equation (5.34) with respect to λ results in

$$D_{\tilde{u}_\mathcal{E}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L})\cdot D_\lambda M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)+B_\mathcal{E}^T=0, \tag{5.42}$$

which is equivalent to

$$D_\lambda M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)=-\left(D_{\tilde{u}_\mathcal{E}}\widetilde{K}_\mathcal{E}(M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda),\tilde{u}_\mathcal{L})\right)^{-1}B_\mathcal{E}^T. \tag{5.43}$$

Recalling that the preconditioner $M(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)$ is the identity in $\tilde{u}_\mathcal{L}$ and λ and using the same notation as in Equation (5.37), the derivative $DM(\tilde{u}_\mathcal{E},\tilde{u}_\mathcal{L},\lambda)$ of the preconditioner M in iteration k is obtained as

$$\begin{aligned}&DM(\tilde{u}_\mathcal{E}^{(k)},\tilde{u}_\mathcal{L}^{(k)},\lambda^{(k)})\\&=\begin{bmatrix}0 & -D\widetilde{K}_{\mathcal{E}\mathcal{E}}(g^{(k)})^{-1}D\widetilde{K}_{\mathcal{E}\mathcal{L}}(g^{(k)}) & -D\widetilde{K}_{\mathcal{E}\mathcal{E}}(g^{(k)})^{-1}B_\mathcal{E}^T\\0 & I & 0\\0 & 0 & I\end{bmatrix}.\end{aligned} \tag{5.44}$$

Thus, the left-hand side of Equation (5.27) writes

$$
\begin{aligned}
&DA(g^{(k)},\lambda^{(k)})\cdot DM(g^{(k)},\lambda^{(k)})\\
&=\begin{bmatrix} D\widetilde{K}_{\mathcal{E}\mathcal{E}} & D\widetilde{K}_{\mathcal{E}\mathcal{L}} & B_{\mathcal{E}}^T \\ D\widetilde{K}_{\mathcal{L}\mathcal{E}} & D\widetilde{K}_{\mathcal{L}\mathcal{L}} & B_{\mathcal{L}}^T \\ B_{\mathcal{E}} & B_{\mathcal{L}} & 0 \end{bmatrix}\cdot\begin{bmatrix} 0 & -D\widetilde{K}_{\mathcal{E}\mathcal{E}}^{-1}D\widetilde{K}_{\mathcal{E}\mathcal{L}} & -D\widetilde{K}_{\mathcal{E}\mathcal{E}}^{-1}B_{\mathcal{E}}^T \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}\\
&=\begin{bmatrix} 0 & 0 & 0 \\ 0 & -D\widetilde{K}_{\mathcal{L}\mathcal{E}}D\widetilde{K}_{\mathcal{E}\mathcal{E}}^{-1}D\widetilde{K}_{\mathcal{E}\mathcal{L}}+D\widetilde{K}_{\mathcal{L}\mathcal{L}} & -D\widetilde{K}_{\mathcal{L}\mathcal{E}}D\widetilde{K}_{\mathcal{E}\mathcal{E}}^{-1}B_{\mathcal{E}}^T+B_{\mathcal{L}}^T \\ 0 & -B_{\mathcal{E}}D\widetilde{K}_{\mathcal{E}\mathcal{E}}^{-1}D\widetilde{K}_{\mathcal{E}\mathcal{L}}+B_{\mathcal{L}} & -B_{\mathcal{E}}D\widetilde{K}_{\mathcal{E}\mathcal{E}}^{-1}B_{\mathcal{E}}^T \end{bmatrix}\\
&=:\begin{bmatrix} 0 & 0 & 0 \\ 0 & DS_{\mathcal{L}\mathcal{L}} & DS_{\mathcal{L}\lambda} \\ 0 & DS_{\lambda\mathcal{L}} & DS_{\lambda\lambda} \end{bmatrix};
\end{aligned}
\tag{5.45}
$$

see [84, Eq. (11)] and [85, Eq. (15)]. The tangent $DA(g^{(k)},\lambda^{(k)})$ is obtained by computing the partial derivatives of the components of the nonlinear system in Equation (5.29); see also Equation (5.48). Note that we have suppressed the function evaluation in $g^{(k)}$ for a better readability. We introduce the operator

$$
DS_{\mathcal{L}}(g^{(k)}) := \begin{bmatrix} DS_{\mathcal{L}\mathcal{L}}(g^{(k)}) & DS_{\mathcal{L}\lambda}(g^{(k)}) \\ DS_{\lambda\mathcal{L}}(g^{(k)}) & DS_{\lambda\lambda}(g^{(k)}) \end{bmatrix}. \tag{5.46}
$$

Under sufficient conditions, i.e., choosing a suitable set of primal variables (see again Section 5.3), the operator $DS_{\mathcal{L}}$, which results from the multiplication of the tangents of the nonlinear system $A(\tilde{u},\lambda)$ and the nonlinear right-preconditioner, is the tangent of the nonlinear Schur complement $S_{\mathcal{L}}$ introduced in Equation (5.33). This result is obtained from the implicit function theorem and the chain rule.

Assuming that we have computed $g^{(k)}$ by solving Equation (5.34), we can write the Newton system (5.27) by using Equations (5.33) and (5.45) as

$$
\begin{bmatrix} 0 & 0 & 0 \\ 0 & DS_{\mathcal{L}\mathcal{L}} & DS_{\mathcal{L}\lambda} \\ 0 & DS_{\lambda\mathcal{L}} & DS_{\lambda\lambda} \end{bmatrix}\begin{bmatrix} \delta\tilde{u}_{\mathcal{E}}^{(k)} \\ \delta\tilde{u}_{\mathcal{L}}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{K}_{\mathcal{L}}(g^{(k)}) + B_{\mathcal{L}}^T\lambda^{(k)} - \tilde{f}_{\mathcal{L}} \\ Bg^{(k)} \end{bmatrix}. \tag{5.47}
$$

As mentioned before, the preconditioner $M(\tilde{u}^{(k)},\lambda^{(k)})$ is independent of $\tilde{u}_{\mathcal{E}}^{(k)}$. Nonetheless, we use $\tilde{u}_{\mathcal{E}}^{(k+1)}$ as an initial guess for the computation of $g^{(k+1)}$; see Figure 5.2. Let us note that in efficient implementations the tangent $DS_{\mathcal{L}}$ of the Schur complement is never computed. Furthermore, the left-hand side of the Schur complement system (5.47) can also be obtained by eliminating

the block $D\widetilde{K}_{\mathcal{E}\mathcal{E}}(g^{(k)})$ from $DA(g^{(k)}, \lambda^{(k)})$. Thus, we replace the left-hand side in Equation (5.47) by

$$DA(g^{(k)}, \lambda^{(k)}) = \begin{bmatrix} D\widetilde{K}_{\mathcal{E}\mathcal{E}}(g^{(k)}) & D\widetilde{K}_{\mathcal{E}\mathcal{L}}(g^{(k)}) & B_{\mathcal{E}}^T \\ D\widetilde{K}_{\mathcal{L}\mathcal{E}}(g^{(k)}) & D\widetilde{K}_{\mathcal{L}\mathcal{L}}(g^{(k)}) & B_{\mathcal{L}}^T \\ B_{\mathcal{E}} & B_{\mathcal{L}} & 0 \end{bmatrix}; \tag{5.48}$$

i.e., we will solve

$$\begin{bmatrix} D\widetilde{K}_{\mathcal{E}\mathcal{E}} & D\widetilde{K}_{\mathcal{E}\mathcal{L}} & B_{\mathcal{E}}^T \\ D\widetilde{K}_{\mathcal{L}\mathcal{E}} & D\widetilde{K}_{\mathcal{L}\mathcal{L}} & B_{\mathcal{L}}^T \\ B_{\mathcal{E}} & B_{\mathcal{L}} & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_{\mathcal{E}}^{(k)} \\ \delta\tilde{u}_{\mathcal{L}}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{K}_{\mathcal{L}} + B_{\mathcal{L}}^T\lambda^{(k)} - \tilde{f}_{\mathcal{L}} \\ Bg^{(k)} \end{bmatrix} \tag{5.49}$$

instead of Equation (5.47); see Section 5.4.2. Again, we have suppressed the function evaluation point $g^{(k)}$ for a better readability. Note that the updates $\delta\tilde{u}_{\mathcal{L}}^{(k)}$ and $\delta\lambda^{(k)}$ are not affected, but we additionally obtain an update $\delta\tilde{u}_{\mathcal{E}}^{(k)}$, which might be useful. Moreover, inexact or inexact reduced FETI-DP methods can be considered when using Equation (5.49); see Section 5.4.4.

There are several publications in which different variants of nonlinear FETI-DP methods were introduced; see e.g. [76, 77, 80, 82–85]. These methods can now be represented by a single algorithm; see Figure 5.2.

5.4.2 Different Variants of Nonlinear FETI-DP

As mentioned before, the application of the nonlinear preconditioner $M(\tilde{u}, \lambda)$ can be interpreted as a partial nonlinear elimination. Hence, the choice of the preconditioner determines the set of variables which will be eliminated and strongly affects the properties of the nonlinear FETI-DP method. In the past, four different choices of the preconditioner M have been considered, which result in four different nonlinear FETI-DP variants.

First, a linear preconditioner $M = I$ was considered; see [76, 77, 80, 82, 83, 85]. Here, no variables are eliminated nonlinearly and this method is called **Nonlinear-FETI-DP-1** or, to shorten the notation, **NL-1**. In contrast to the linear preconditioner, nonlinear preconditioning of the variable $\tilde{u}$ was considered later; see [77,85]. Here, the variable $\tilde{u}$ is eliminated nonlinearly and the method is called **Nonlinear-FETI-DP-2**, or, in short, **NL-2**. Motivated by the variable splitting in FETI-DP methods, two further nonlinear FETI-DP variants were introduced, which can be characterized by a nonlinear preconditioning of u_B and u_I, respectively. Both methods were described in [84,85]. Precondition-

Init: $\tilde{u}^{(0)} \in \widetilde{W}$, $\lambda^{(0)} \in V$
for $k = 0, \ldots,$ convergence **do**
  Compute:
  $\left(g^{(k)}, \lambda^{(k)}\right) := \left(M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)}), \lambda^{(k)}\right) = M(\tilde{u}^{(k)}, \lambda^{(k)})$
  /* Often requires solution of localized nonlinear problems; see Equation (5.36) and Figure 5.7 (top left)*/
  if $||A(g^{(k)}, \lambda^{(k)})||$ sufficiently small **then**
    break //Convergence of nonlinear FETI-DP
  end if

  Solve (with Krylov subspace method):
  $\left(DA(g^{(k)}, \lambda^{(k)}) \cdot DM(\tilde{u}^{(k)}, \lambda^{(k)})\right) \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = A(g^{(k)}, \lambda^{(k)})$
  Update $\tilde{u}$: $\tilde{u}^{(k+1)} = \tilde{u}^{(k)} - \alpha^{(k)}\delta\tilde{u}^{(k)}$
  Update λ: $\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}\delta\lambda^{(k)}$
end for

Figure 5.2: Nonlinear FETI-DP algorithm(s). We always use $\tilde{u}^{(k+1)}$ as initial value for the computation of $g^{(k+1)}$. Note that $||A(g^{(k)}, \lambda^{(k)})||$ and $||A(\tilde{u}^{(k+1)}, \lambda^{(k+1)})|$ can be replaced by $||\delta\tilde{u}_{\mathcal{E}}^{(k)}||$ and $||\delta\tilde{u}^{(k)}||$, respectively. This figure has been similarly published in [85, Fig. 3].

ing in u_B leads to **Nonlinear-FETI-DP-3** or **NL-3** and preconditioning in u_I leads to **Nonlinear-FETI-DP-4** or **NL-4**. The four different methods are considered in the following sections in more detail; see Sections 5.4.2.1 to 5.4.2.4 and also Figure 5.3.

The choices of $\mathcal{E}$ and $\mathcal{L}$ are not restricted to these cases. Therefore, we have recently implemented a nonlinear FETI-DP method which chooses a problem-dependent elimination set based on information of the nonlinear residual; see Section 5.4.2.5. It is called **Nonlinear-FETI-DP-Res** and we will refer to it as **NL-Res** in the remainder of this thesis.

Let us note that all nonlinear FETI-DP methods as well as the Newton-Krylov-FETI-DP approach reduce to the standard FETI-DP method and are therefore equivalent when they are applied to a linear problem.

5.4.2.1 Nonlinear-FETI-DP-1

We start with the NL-1 method, which is one of the extreme cases of nonlinear partial elimination.

To obtain NL-1 as introduced in, e.g., [76, 77], the index set $\mathcal{E}$ is chosen to be the empty set, i.e., no variables are nonlinearly eliminated and $\mathcal{E} := \emptyset$ holds.

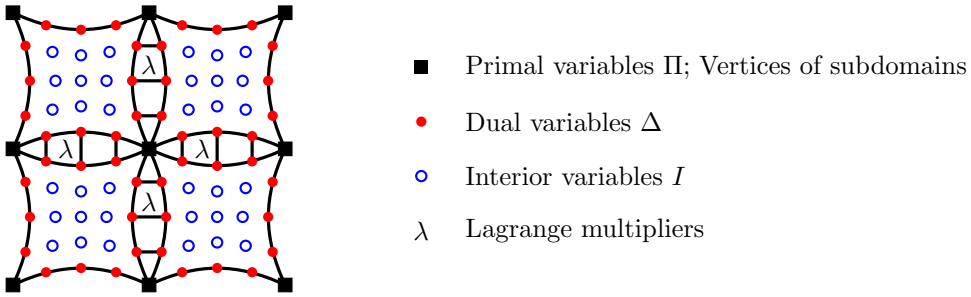


Figure 5.3: Coupling of nonlinear local problems in the primal variables Π (black squares). Continuity in the dual interface variables Δ (red dots) is enforced by Lagrange multipliers λ. The remaining interior variables I are represented by blue circles. The nonlinear preconditioning in NL-4 performs exclusively on the interior variables (blue circles). In NL-3, the nonlinear preconditioning additionally operates on the dual interface variables (red dots). For NL-2 the nonlinear preconditioning operates on all variables, which includes the primal variables (black squares). This figure has already been published in [85]; see [85, Fig. 2].

As a consequence, we have $\mathcal{L} := \left[I \; \Delta \; \Pi \right]$. Thus, the preconditioner $M(\tilde{u}, \lambda)$ reduces to the identity, i.e.,

$$M(\tilde{u}, \lambda) := (\tilde{u}, \lambda). \tag{5.50}$$

Again, using the notation $g^{(k)} := M_{\tilde{u}}(\tilde{u}, \lambda)$ (see Equation (5.35)), we obtain $g^{(k)} = \tilde{u}^{(k)}$. The preconditioner $M(\tilde{u}, \lambda)$ automatically yields $DM(\tilde{u}, \lambda) = I$. Therefore, the linearized system (5.27) writes in a compressed form

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f} \\ B\tilde{u}^{(k)} \end{bmatrix}, \tag{5.51}$$

which reduces to the dual Schur complement system

$$\begin{aligned} &B \left(D\widetilde{K}(\tilde{u}^{(k)}) \right)^{-1} B^T \delta\lambda^{(k)} \\ &\quad = -B\tilde{u}^{(k)} + B \left(D\widetilde{K}(\tilde{u}^{(k)})^{-1} \right) \left(\widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f} \right), \end{aligned} \tag{5.52}$$

as is standard in exact or inexact FETI-DP methods. Thus, we can solve Equation (5.52) using a preconditioned Krylov subspace method and a suitable choice of the preconditioner is the Dirichlet preconditioner; see [41,150] and also Equation (5.12).

Let us note that $D\widetilde{K}(\tilde{u}^{(k)})^{-1}$ is never build explicitly, but its application to a vector is computed by local, i.e., parallel, sparse LU-factorizations and the solution of the globally coupled coarse problem, i.e., a small Schur complement system; see Equations (5.5) and (5.6) and [77] as well as Section 5.2.

For our model problems, it turns out that NL-1 performs similar to the classic NK-FETI-DP method, which will be introduced in Section 5.5. To improve the performance of NL-1, we can compute an initial value by solving the nonlinear equation

$$\widetilde{K}(\tilde{u}^{(0)}) + B^T\lambda^{(0)} - \tilde{f} = 0 \tag{5.53}$$

for given initial values $\tilde{u}^{(0)}$ and $\lambda^{(0)}$; see [76,77,80,83]. Throughout this thesis, $\lambda^{(0)} = 0$ is chosen and most numerical results for NL-1 are obtained by including the computation of the initial value $\tilde{u}^{(0)}$.

5.4.2.2 Nonlinear-FETI-DP-2

The other extreme case of nonlinear partial elimination is the NL-2 method, where the variable $\tilde{u}$ is completely eliminated. We obtain the NL-2 method as described in [76,77] by choosing $\mathcal{E} := \begin{bmatrix} I & \Delta & \Pi \end{bmatrix}$ and $\mathcal{L} := \emptyset$, which is vice versa compared to NL-1; see Section 5.4.2.1. Since the index set $\mathcal{E}$ contains all degrees of freedom, we obtain $M_{u_\mathcal{E}}(\tilde{u}_\mathcal{E}, \tilde{u}_\mathcal{L}, \lambda) = M_{\tilde{u}}(\tilde{u}, \lambda)$ and the preconditioner is defined implicitly by

$$\widetilde{K}(M_{\tilde{u}}(\tilde{u}, \lambda)) + B^T\lambda - \tilde{f} = 0;$$

see also Equation (5.34). The solution $g^{(k)} := M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)})$ (see Equation (5.35)) is achieved by the application of Newton's method. It yields the Newton iteration

$$g_{l+1}^{(k)} := g_l^{(k)} - \left(D\widetilde{K}(g_l^{(k)})\right)^{-1} \left(\widetilde{K}(g_l^{(k)}) + B^T\lambda^{(k)} - \tilde{f}\right), \tag{5.54}$$

which is assumed to converge to the solution $g^{(k)}$; see also Equation (5.36). As before, k represents the outer Newton iteration and l represents the inner Newton iteration, which is needed for the computation of $g^{(k)}$, which again is needed for the right-hand side of the linearized system; cf. Equation (5.27). For NL-2, the resulting system writes in compressed form

$$\begin{bmatrix} D\widetilde{K}(g^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ Bg^{(k)} \end{bmatrix}. \tag{5.55}$$

Again, the resulting Schur complement system

$$B\left(D\widetilde{K}(g^{(k)})\right)^{-1}B^T\delta\lambda^{(k)} = -Bg^{(k)} \tag{5.56}$$

can be solved by a Krylov subspace method and a suitable preconditioner such as the Dirichlet preconditioner (see [41, 150] and Equation (5.12)). By solving Equation (5.56) for $\delta\lambda^{(k)}$, we automatically assume that the Newton update $\delta\tilde{u}^{(k)}$ is zero.

As mentioned at the end of Section 5.4.1, it is more practical to solve Equation (5.55) instead of Equation (5.56). Since Equation (5.56) is obtained from Equation (5.55), this does not change the solution $\delta\lambda^{(k)}$, but we additionally obtain $\delta\tilde{u}^{(k)}$, which can be used to update the initial value for the computation of $g^{(k+1)}$, i.e., it accelerates the computation of $g^{(k+1)}$ by choosing a better initial value. Furthermore, it is possible to apply inexact or inexact reduced FETI-DP methods without changing the solution; see Section 5.4.4.

The NL-2 approach can be characterized by an exact nonlinear elimination of the variable $\tilde{u}$. We have an inner and an outer Newton iteration, where the inner Newton iteration represents the computation of the right-hand side of the linearized system which has to be solved in the outer Newton iteration.

Note that in NL-3 and NL-4, we also have inner and outer Newton iterations, but the elimination sets do not contain the primal variables. Hence, the inner Newton iteration in NL-3 and NL-4 is expected to be cheaper compared to NL-2, since it does not contain the coarse problem and its all-to-all communication.

5.4.2.3 Nonlinear-FETI-DP-3

The next two nonlinear FETI-DP methods are motivated by the variable splitting of the classical FETI-DP approach. The idea is to design methods with completely local nonlinear Newton iterations to improve scalability. Therefore, we have to ensure that the elimination set $\mathcal{E}$ does not contain the primal variables. The NL-3 approach, as described in [84], is obtained by choosing $\mathcal{E} := B$ and $\mathcal{L} := \Pi$. As before, the preconditioner $M_{\tilde{u}_\mathcal{E}}(\tilde{u}_\mathcal{E}, \tilde{u}_\mathcal{L}, \lambda)$ is implicitly defined by Equation (5.34); see also Section 5.4.2.2. From the choice of $\mathcal{E}$ and $\mathcal{L}$, we obtain $g^{(k)} = (g_B^{(k)}, \tilde{u}_\Pi)$; see Equation (5.38). Applying Newton's method to Equation (5.34) yields the iteration

$$g_{B,l+1}^{(k)} := g_{B,l}^{(k)} - \left(D\widetilde{K}_{BB}(g_{B,l}^{(k)}, \tilde{u}_\Pi^{(k)})\right)^{-1}\left(\widetilde{K}_B(g_{B,l}^{(k)}, \tilde{u}_\Pi^{(k)}) + B_B^T\lambda^{(k)} - \tilde{f}_B\right), \tag{5.57}$$

which converges to the solution $g_B^{(k)}$; see Equation (5.36).

By definition, we have continuity in all primal variables Π. Hence, we obtain $B_{\mathcal{L}} := B_{\Pi} = 0$. Then, the linearized system writes

$$\begin{bmatrix} D\widetilde{K}_{BB}(g^{(k)}) & D\widetilde{K}_{B\Pi}(g^{(k)}) & B_B^T \\ D\widetilde{K}_{\Pi B}(g^{(k)}) & D\widetilde{K}_{\Pi\Pi}(g^{(k)}) & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_B^{(k)} \\ \delta\tilde{u}_{\Pi}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{K}_{\Pi}(g^{(k)}) - \tilde{f}_{\Pi} \\ Bg^{(k)} \end{bmatrix}. \tag{5.58}$$

In the inner Newton iteration of NL-3, local nonlinear problems in the variable $\tilde{u}_B$ have to be solved, which can be computed completely independent for each subdomain and do not include the solution of any FETI-DP coarse problem. In comparison to the NL-2 approach, this method dramatically reduces the number of primal assembly processes and FETI-DP coarse solves since this has to be done only in the outer Newton iteration. This is expected to improve the scalability.

As mentioned before, the inner Newton iteration in NL-3 does not require any synchronization. Consequently, the local nonlinear problems can be solved completely in parallel. It is obvious that different subdomains may need different numbers of inner Newton iterations. Assuming that we have assigned each subdomain to a single core, this can lead to problem-dependent load imbalances. So far, dynamic load balancing, e.g., by resizing the subdomains, is currently not feasible; see [88]. Instead, we set those cores to sleep, which have already finished the inner Newton iteration. This approach can save a significant amount of energy. For details, we refer to [88].

5.4.2.4 Nonlinear-FETI-DP-4

The NL-4 approach was first introduced in [84]. It is quite similar to the NL-3 approach; see Section 5.4.2.3. We choose $\mathcal{E} := I$ and $\mathcal{L} := \begin{bmatrix} \Delta & \Pi \end{bmatrix} = \Gamma$. Again, the preconditioner $M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{E}}, \tilde{u}_{\mathcal{L}}, \lambda)$ is implicitly defined by Equation (5.34); see also Section 5.4.2.2. We obtain $g^{(k)} = (g_I^{(k)}, \tilde{u}_{\Gamma})$ from Equation (5.38) and the application of Newton's method to Equation (5.34) yields the iteration

$$g_{I,l+1}^{(k)} := g_{I,l}^{(k)} - \left(D\widetilde{K}_{II}(g_{I,l}^{(k)}, \tilde{u}_{\Gamma}^{(k)}) \right)^{-1} \left(\widetilde{K}_I(g_{I,l}^{(k)}, \tilde{u}_{\Gamma}^{(k)}) - \tilde{f}_I \right), \tag{5.59}$$

which converges to the solution $g_I^{(k)}$; see Equation (5.36). Then, the linearized system writes

$$\begin{bmatrix} D\widetilde{K}_{II}(g^{(k)}) & D\widetilde{K}_{I\Gamma}(g^{(k)}) & 0 \\ D\widetilde{K}_{\Gamma I}(g^{(k)}) & D\widetilde{K}_{\Gamma\Gamma}(g^{(k)}) & B_{\Gamma}^T \\ 0 & B_{\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_I^{(k)} \\ \delta\tilde{u}_{\Gamma}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{K}_{\Gamma}(g^{(k)}) + B_{\Gamma}^T\lambda^{(k)} - \tilde{f}_{\Gamma} \\ Bg^{(k)} \end{bmatrix}. \tag{5.60}$$

Analogously to NL-3, no coarse problem has to be solved in the inner Newton iteration, which results in completely local nonlinear problems. Here, only the inner variables $\tilde{u}_I$ are eliminated, which results in cheaper local solves compared to NL-3. Again, due to the local nonlinear problems in the inner Newton iteration, problem-dependent load imbalances can be used to save energy as in NL-3; see Section 5.4.2.3 and [88].

5.4.2.5 Choosing Problem-Dependent Elimination Sets in Nonlinear FETI-DP

The framework of nonlinear FETI-DP methods also allows to choose arbitrary index sets $\mathcal{E}$ and $\mathcal{L}$, which results in problem-dependent nonlinear FETI-DP methods. The goal is to select the index set $\mathcal{E}$ in such a way that the preconditioner M is efficient and at the same time $\mathcal{E}$ is as small as possible in order to keep the effort of the inner Newton iteration low. Of course, the resulting nonlinear problems are usually no longer completely independent as it is the case for NL-3 and NL-4 since the elimination set might also contain primal variables. Nonetheless, we have usually less coupling compared to NL-2.

For the choice of a problem-dependent elimination set $\mathcal{E}$, we use a strategy introduced by Gong and Cai in [50] in the context of inexact Newton methods. It is recommended that the index set $\mathcal{E}$ contains the degrees of freedom which have a high absolute residual value.

As before, we consider the nonlinear problem

$$A(M(\tilde{u}, \lambda)) = 0;$$

cf. Equation (5.25). Generally, we are interested in the solution $\tilde{u}$ of the nonlinear system at hand. The Lagrange multipliers are only introduced to fulfill the continuity requirement of the final solution $\tilde{u}$. Therefore, it is reasonable to consider the nonlinear residual $A(\tilde{u}, \lambda)$ restricted to the variable $\tilde{u}$. Assuming that we have computed the solution $\lambda_*^{(k)}$ and $\tilde{u}_*^{(k)}$ at the end of iteration k (see Section 5.4), we obtain

$$A(\tilde{u}_*^{(k)}, \lambda_*^{(k)})\big|_{\tilde{u}} = \widetilde{K}(\tilde{u}_*^{(k)}) + B^T \lambda_*^{(k)} - \tilde{f}.$$

Since $\widetilde{K}(\tilde{u}_*^{(k)})$ and $\tilde{f}$ are only assembled in the primal variables, other interface nodes usually have different values in the subdomains they belong to. Therefore, we consider the fully assembled residual

$$\overline{r}^{(k)} := R^T \cdot A(\tilde{u}_*^{(k)}, \lambda_*^{(k)})\big|_{\tilde{u}},$$

i.e., we apply the assembly operator R^T (see Sections 5.2 and 5.6.1) to $A(\tilde{u}_*^{(k)}, \lambda_*^{(k)})\big|_{\tilde{u}}$. Let us introduce the following notation. Depending on the number d of degrees of freedom per finite element node, we have d associated residual values per finite element node. Let

$$\mathcal{S} = \{1, \ldots, n\}$$

be the index set of all finite element nodes and

$$\mathcal{I} = \{1, \ldots, m\}$$

the index set of all degrees of freedom corresponding to the variable $\tilde{u}$, where $m = n \cdot d$ holds. The residual vector $\overline{r}^{(k)}$ decomposes into n subvectors $\overline{r}_{(i)}^{(k)} \in \mathbb{R}^d$, where each $\overline{r}_{(i)}^{(k)}$ holds the residual values $\overline{r}_{(i)_j}^{(k)}$, $j = 1, \ldots, d$, corresponding to the degrees of freedom of finite element node i.

The maximum absolute residual value of the current iteration k is given by the infinity norm $||\overline{r}^{(k)}||_\infty$. With the introduced notation, we obtain

$$\begin{aligned} ||\overline{r}^{(k)}||_\infty = \max_{i=1}^{n} ||\overline{r}_{(i)}^{(k)}||_\infty &= \max_{i=1}^{n} \left(\max_{j=1}^{d} |\overline{r}_{(i)_j}^{(k)}| \right) \\ &= \max_{i=1}^{n} \left(\max_{j=1}^{d} |R_{(i)_j}^T A(\tilde{u}_*^{(k)}, \lambda_*^{(k)})\big|_{\tilde{u}}| \right), \end{aligned}$$

where $R_{(i)_j}^T$ is the row of R^T corresponding to the degree of freedom i_j. Similar to [50], we want to determine an index set of degrees of freedom that has to be eliminated nonlinearly. Therefore, we introduce the index set

$$\mathcal{S}_b^{(k)} := \left\{ i \in \mathcal{S} \;\middle|\; ||\overline{r}_{(i)}^{(k)}||_\infty \geq \rho_{\text{res}} \cdot ||\overline{r}^{(k)}||_\infty \right\} \subseteq \mathcal{S},$$

which contains all indices of finite element nodes with at least one degree of freedom with a large residual component. Here, $\rho_{\text{res}} \in (0, 1]$ defines the tolerance to distinguish wether a finite element node lies within $S_b^{(k)}$ or not. The smaller ρ_{res}, the more finite element nodes are contained in the index set $S_b^{(k)}$. Furthermore, we introduce the index set

$$\mathcal{I}_b^{(k)} := \left\{ i_j \in \mathcal{I} \;\middle|\; i \in \mathcal{S}_b^{(k)},\ j = 1, \ldots, d \right\} \subseteq \mathcal{I},$$

i.e., ρ_{res} also affects the number of degrees of freedom in $\mathcal{I}_b^{(k)}$. In contrast to [50], we do not consider the degrees of freedom of a single finite element node separately, i.e., we decide to eliminate all degrees of freedom corresponding to a finite element node if at least one degree of freedom has a higher residual

component than the maximum residual value multiplied with ρ_{res}. Thus, the elimination set usually contains degrees of freedom with residual values smaller than $\rho_{\text{res}} \cdot ||\overline{r}^{(k)}||_\infty$. Furthermore, we decide to include finite element nodes into $\mathcal{S}_b^{(k)}$ if one corresponding degree of freedom has a residual value as large as $\rho_{\text{res}} \cdot ||\overline{r}^{(k)}||_\infty$, which also enlarges the number of degrees of freedom to be eliminated compared to [50]. Therefore, also the choice $\rho_{\text{res}} = 1$ is reasonable since $S_b^{(k)}$ contains at least one index in this case.

In a last step, we extend the index set $S_b^{(k)}$ using an approach similar to the overlap $\delta \in \mathbb{N}$ in overlapping domain decomposition methods such as the additive overlapping Schwarz method. The index set $S_{b,\delta}^{(k)}$ is defined as

$$\mathcal{S}_{b,\delta}^{(k)} := \mathcal{S}_b^{(k)} \bigcup \left\{ i \in \mathcal{S} \setminus \mathcal{S}_b^{(k)} \;\middle|\; \exists\, l \in \mathcal{S}_b^{(k)} : i \text{ is in the overlap } \delta \text{ of } l \right\} \subseteq \mathcal{S},$$

which is used to remedy sharp jumps in the residual function as is mentioned in [50]. As before, we also introduce the corresponding set of degrees of freedom as

$$\mathcal{I}_{b,\delta}^{(k)} := \left\{ i_j \in \mathcal{I} \;\middle|\; i \in \mathcal{S}_{b,\delta}^{(k)},\ j = 1, \ldots, d \right\} \subseteq \mathcal{I}.$$

Finally, we eliminate all degrees of freedom that are contained in $\mathcal{I}_{b,\delta}^{(k)}$, i.e., we set $\mathcal{E} := \mathcal{I}_{b,\delta}^{(k)}$. We provide numerical results with different choices of δ and ρ_{res}; see Section 6.7.

Note that in our opinion, it would also be possible to choose the Euclidean norm instead of the infinity norm. Certainly, this requires another choice of the parameter ρ_{res}, since the infinity norm is invariant with respect to the length of any vector while the Euclidean norm is not.

Let us give some further remarks regarding the elimination set $\mathcal{E}$. So far, we have considered constant elimination sets throughout the computation, but the framework only requires a constant elimination set $\mathcal{E}$ within the inner Newton iteration. Therefore, in the context of problem-dependent index sets $\mathcal{E}$ and $\mathcal{L}$, it might be advantageous to determine a new elimination set $\mathcal{E}$ at the end of each outer Newton iteration. This strategy is referred to as dynamic computation of the elimination set (see Figure 5.4) and, for the Nonlinear-FETI-DP-Res approach, it is activated for all simulation results presented within this thesis.

The usage of a dynamically computed elimination set also enables the possibility to choose an empty elimination set, i.e., $\mathcal{E} = \varnothing$, which might be advantageous if the current iterate is in the neighborhood of the solution, since Newton's method is expected to converge quadratically in this case. Thus, the nonlinear elimination of some variables might be superfluous or even counterpro-

ductive. Therefore, we perform an NL-1 step ($\mathcal{E} = \varnothing$), whenever the considered norm ($||A(\tilde{u}^{(k)}, \lambda^{(k)})||$ or $||\delta\tilde{u}^{(k)}||$) is smaller than a pre-chosen tolerance tol; see Figure 5.4.

Finally, we have to consider the very first iteration. Assuming that the problem at hand is unknown, it might be disadvantageous to choose an elimination set based on the initial guess, since it provides too little information. In this case, it is possible to start with an empty elimination set, i.e., we perform an NL-1 step and compute the elimination set at the end of the first iteration. This strategy is used for the 2D contact simulations but not for the simulations with the p-Laplace equation; see Section 6.7.

As mentioned before, we refer to the nonlinear FETI-DP approach using a residual-based strategy to determine the elimination set $\mathcal{E}$ as Nonlinear-FETI-DP-Res or, to shorten the notation, NL-Res in the remainder of this thesis.

5.4.3 Remarks on the Preconditioners

In this section, we shortly discuss the properties of the nonlinear preconditioner M and give some remarks on the preconditioners that were introduced in the previous Sections 5.4.2.1 to 5.4.2.5. It is based on the discussions in [85, Sec. 2.5.4] and [85, Sec. 2.5.9].

In general, the preconditioner M fulfills the first two properties listed in Figure 5.1 (left), by definition. Furthermore, M can be computed easier compared to A, since the constraint $Bu = 0$ is omitted. Therefore, the third condition of Figure 5.1 (left) is also satisfied.

As mentioned before, our methods can be seen as nonlinear right-preconditioned Newton methods, which follows from Equation (5.25) and [18]. In right-preconditioned Newton-Krylov methods, the computation of DM^{-1} can be avoided by a first order approximation of M; see [18]. However, we do not use an approximation but compute the tangent exactly; see Equation (5.27) and [85]. Nonetheless, the action of M^{-1} may be approximated under some circumstances; see Section 5.7.

Comparing the preconditioners of NL-1, NL-2, NL-3, NL-4, and NL-Res, it is obvious that the preconditioner of NL-1, which is the identity, is the cheapest one, but it does not give a good approximation to A. However, we can overcome this drawback by the computation of an initial value $\tilde{u}^{(0)}$ from the nonlinear problem $\widetilde{K}(\tilde{u}^{(0)}) = \tilde{f} - B^T\lambda^{(0)}$; see Equation (5.53). Therefore, Newton's method is used and $\lambda^{(0)}$ is a given initial value chosen to be zero throughout this thesis, which is a common choice; see, e.g., [77]. The computation of the initial value also includes a nonlinear coarse problem. Hence, the re-

Init: $\tilde{u}^{(0)} \in \widetilde{W}$, $\lambda^{(0)} \in V$, tol $\in \mathbb{R}$, $\rho_{\text{res}} \in [0,1]$, $\delta \in \mathbb{N}$
Compute $\mathcal{E}$ based on $A(\tilde{u}^{(0)}, \lambda^{(0)})\big|_{\tilde{u}}$ or **choose** $\mathcal{E} = \varnothing$, **depending on** $\tilde{\boldsymbol{u}}^{(\mathbf{0})}$

for $k = 0, \ldots,$ convergence **do**
  Compute:
  $\left(g^{(k)}, \lambda^{(k)}\right) := \left(M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)}), \lambda^{(k)}\right) = M(\tilde{u}^{(k)}, \lambda^{(k)})$
  /* Often requires solution of localized nonlinear problems; see Equation (5.36) and Figure 5.7 (top left)*/
  if $||A(g^{(k)}, \lambda^{(k)})||$ sufficiently small **then**
    break //Convergence of nonlinear FETI-DP
  end if

  Solve (with Krylov subspace method):
  $\left(DA(g^{(k)}, \lambda^{(k)}) \cdot DM(\tilde{u}^{(k)}, \lambda^{(k)})\right) \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = A(g^{(k)}, \lambda^{(k)})$
  Update $\tilde{\boldsymbol{u}}$**:** $\tilde{u}^{(k+1)} = \tilde{u}^{(k)} - \alpha^{(k)}\delta\tilde{u}^{(k)}$
  Update $\boldsymbol{\lambda}$**:** $\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}\delta\lambda^{(k)}$
  if $||A(\tilde{u}^{(k+1)}, \lambda^{(k+1)})|| \leq$ tol **then**
    Set $\mathcal{E} = \varnothing$
  else
    if dynamic computation of $\mathcal{E}$ activated **then**
      Compute $\mathcal{E}$ based on $A(\tilde{u}^{(k+1)}, \lambda^{(k+1)})\big|_{\tilde{u}}$, ρ_{res}, and δ
    else
      Reuse $\mathcal{E}$ from the last iteration
    end if
  end if
end for

Figure 5.4: Nonlinear-FETI-DP-Res algorithm using a problem-dependent choice of the elimination set $\mathcal{E}$ based on the residual values. We always use $\tilde{u}^{(k+1)}$ as the initial value for the computation of $g^{(k+1)}$. Note that $||A(g^{(k)}, \lambda^{(k)})||$ and $||A(\tilde{u}^{(k+1)}, \lambda^{(k+1)})|$ can be replaced by $||\delta\tilde{u}_{\mathcal{E}}^{(k)}||$ and $||\delta\tilde{u}^{(k)}||$, respectively.

sulting costs of the computation of the initial value in NL-1 are similar to the costs of the application of the preconditioner M in NL-2, which is the most expensive preconditioner of all nonlinear FETI-DP variants. Assuming that a good coarse space is chosen, M will be a good approximation of the inverse of A and, therefore, M will be a good preconditioner for A. As mentioned before, the preconditioners in NL-3 and NL-4 do not include a nonlinear coarse problem. Thus, the computation of M is computationally cheaper compared to NL-2. Furthermore, the computation only includes local solves and can be done completely in parallel. By construction, the preconditioner in NL-4 is cheaper

compared to that in NL-3, but it is only a good preconditioner for A, if the subdomains are chosen such that the nonlinearities of the problem do not touch the interface; cf. the numerical results in Chapter 6.

Due to the problem-dependent choice of the elimination set in NL-Res, the application of the preconditioner is usually not completely parallel but only requires the solution of a subproblem of the FETI-DP coarse problem. In NL-2, NL-3, and NL-4, the elimination set contains the inner variables and, therefore, most variables are eliminated. This is different for the NL-Res approach. The elimination set only contains the most critical degrees of freedom depending on the choice of ρ_{res}. As a consequence, the elimination set is much smaller compared to NL-2, NL-3, and NL-4, even if a large δ environment (see Section 5.4.2.5) is chosen; see also Section 6.7. Thus, the application of the preconditioner of NL-Res operates on less degrees of freedom compared to NL-2, NL-3, and NL-4, but it also requires the solution of a subproblem of the coarse problem. Consequently, the application of the preconditioner in NL-Res might be cheaper compared to NL-3 and NL-4 whenever the additional costs of the global subproblem are insignificant. However, the preconditioner in NL-Res is probably more effective compared to NL-3 and NL-4 since it contains some relevant primal variables that are not affected by the preconditioners in NL-3 and NL-4.

5.4.4 Using Algebraic Multigrid to Approximate the Coarse Problem of Nonlinear FETI-DP Methods

In this section, we briefly discuss how we achieve a good parallel scalability for nonlinear FETI-DP methods on very large scales by combining our nonlinear FETI-DP methods with the framework of inexact reduced FETI-DP. We follow the discussions in [85, Sec. 2.6] and [107, Sec. 3].

It is well known that the exact solution of the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$ is a scaling bottleneck for large coarse problems; see, e.g., [94, 107]. Usually, the coarse problem grows with the number of subdomains, e.g., if subdomain vertices are chosen as primal variables. If we assume that we have one subdomain per compute core, it automatically follows that the exact solution of the FETI-DP coarse problem is a scaling bottleneck at large scales. To overcome this issue, we want to solve the coarse problem approximately using an algebraic multigrid (AMG) method. Note that solving the coarse problem of standard (linear or nonlinear) FETI-DP methods inexactly, e.g., with a multilevel solver, perturbs the solution since the coarse problem is part of the operator and not part of the preconditioner. To use (linear or nonlinear) FETI-DP methods on

the largest supercomputers efficiently without perturbing the solution, a different approach was introduced in [80, 94], where the solution of the coarse problem is transferred to the preconditioner.

Recalling Equation (5.49) from Section 5.4.1, all nonlinear FETI-DP variants result in a linearized system of the form

$$\begin{bmatrix} D\widetilde{K}_{BB}(\tilde{u}^{(k)}) & D\widetilde{K}_{B\Pi}(\tilde{u}^{(k)}) & B_B^T \\ D\widetilde{K}_{\Pi B}(\tilde{u}^{(k)}) & D\widetilde{K}_{\Pi\Pi}(\tilde{u}^{(k)}) & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_B \\ \delta\tilde{u}_\Pi \\ \delta\lambda \end{bmatrix} = \mathrm{rhs}_1.$$

Note that the left-hand side is identical for all nonlinear FETI-DP methods, but the right-hand side differs due to the different choices of the elimination sets; see Equation (5.49) as well as Equations (5.51), (5.55), (5.58) and (5.60). The idea is to construct FETI-DP methods that perform a single iteration of a multilevel preconditioner instead of an exact factorization of the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$; see Equation (5.6). For this purpose, we perform a block elimination of $\tilde{u}_B$, as presented in [92], which leads to

$$\begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -(D\widetilde{K}_{\Pi B})(D\widetilde{K}_{BB})^{-1}B_B^T \\ -B_B(D\widetilde{K}_{BB})^{-1}(D\widetilde{K}_{B\Pi}) & -B_B(D\widetilde{K}_{BB})^{-1}B_B^T \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_\Pi \\ \delta\lambda \end{bmatrix} = \mathrm{rhs}_2,$$

where we have suppressed the function evaluation point $\tilde{u}$ for a better readability. Again, rhs_2 differs for the different nonlinear methods described in Sections 5.4.2.1 to 5.4.2.5. Afterwards, we use a block triangular preconditioner for saddle point systems in combination with GMRES; see [80, 92, 94]. Throughout this thesis, a single iteration of BoomerAMG [59] is used as a preconditioner for the $\widetilde{S}_{\Pi\Pi}$ block and, as in [92], the standard FETI-DP Dirichlet preconditioner (see [41, 150] and Equation (5.12)) is used as a preconditioner for the lower right block; see [92]. Let us note that it is also possible to use the conjugate gradient (CG) method as a Krylov subspace method when using the well-known symmetric positive definite reformulation of the achieved saddle point system; see [15, 73, 92]. As was shown in [2], BoomerAMG is parallel scalable for linear elasticity problems on more than half a million cores when using appropriate interpolation strategies from [3] and, therefore, the usage of BoomerAMG provides a substantial improvement of the scalability of FETI-DP methods.

For further details regarding inexact reduced FETI-DP methods as well as inexact FETI-DP methods, we refer to [92].

5.4.5 Energy Efficiency in NL-Res and NL-2

For the example of NL-3, a more energy efficient implementation was introduced in [88]. It takes advantage of the fact that the subdomain problems can be completely solved in parallel for NL-3, which probably leads to load imbalances. To save energy, all cores belonging to subdomains that have already finished the local solution process are set so sleep. For further details, we refer to [88].

Note that the strategy presented in [88] is not restricted to NL-3, but can be applied for all preconditioners with a nonempty elimination set that do not contain any primal variables. Therefore, it also covers NL-4 and NL-Res in case that no primal variable is included into the elimination set. Since the latter is usually not the case, let us give some remarks on the energy efficiency of NL-Res and NL-2.

First, let us note that the primal variables $\Pi_{\mathcal{E}}$, which belong to the elimination set $\mathcal{E}$, are usually a (small) subset of all primal variables. Accordingly, the nonlinear elimination of the primal variables $\Pi_{\mathcal{E}}$ requires the solution of a subproblem of the global coarse problem. Therefore, we need the local stiffness matrices and right-hand sides of all neighboring subdomains. Consequently, the elimination of a primal variable automatically involves all subdomains that are adjacent to primal variables in $\Pi_{\mathcal{E}}$.

In order to apply a strategy similar to that presented in [88] for the NL-3 approach, we introduce the notation of clusters of subdomains. We consider independent clusters of subdomains, i.e., each subdomain exclusively belongs to one cluster.

First of all, each subdomain that is not adjacent to a primal variable in $\Pi_{\mathcal{E}}$ builds its own cluster, i.e., the cluster contains only a single subdomain. The remaining subdomains have to be distributed to different clusters. Two subdomains Ω_i and Ω_j belong to the same cluster $\mathcal{C}$, if there is a path from Ω_i to Ω_j by only crossing primal variables that belong to $\Pi_{\mathcal{E}}$. For sure, all subdomains Ω_k that were passed through on this way also belong to the cluster $\mathcal{C}$. Accordingly, if there is no such path for a subdomain Ω_i with all subdomains that belong to the cluster $\mathcal{C}$, it belongs to a different cluster.

Finally, we can apply the same strategy to NL-Res as introduced in [88], but we have to consider clusters instead of subdomains. Note that each subdomain builds its own cluster if the elimination set does not contain any primal variable. In this case, the notations of clusters and subdomains are equivalent and, therefore, our strategy is exactly the same as in [88].

In FETI-DP methods, usually each subdomain contains at least on primal variable on its interface. For example, we have chosen the subdomain corners

as primal variables throughout this thesis. Recalling that all primal variables belong to the elimination set for NL-2, i.e., we obtain $\Pi_{\mathcal{E}} = \Pi$, it immediately follows that we end up with only one cluster that contains all subdomains. Accordingly, all subdomains are involved throughout the complete nonlinear elimination process and therefore, no subdomain finishes the inner iteration before others.

At the end of this section, we briefly discuss a possible parallel implementation of the energy-efficient NL-Res approach. Usually, each subdomain has its own compute core. For each cluster $\mathcal{C}_l$, $l \in \mathbb{N}$, we introduce an independent subcommunicator that contains the corresponding cores of all subdomains that belong to $\mathcal{C}_l$. In order to safe energy, all cores can be set to sleep, which belong to clusters that have already finished the inner Newton iteration and are waiting for other clusters. Note that the subcommunicators have to be reset at the end of each outer Newton iteration since the elimination set can change; see Section 5.4.2.5. However, it remains constant within the elimination process, i.e., the inner loop.

5.5 Newton-Krylov-FETI-DP

For the sake of completeness, we also describe the traditional approach to tackle nonlinear problems with the FETI-DP method. It is based on the discussion in [107, Sec. 2.2].

Instead of performing a nonlinear domain decomposition, the nonlinear problem is first linearized and afterwards decomposed into subdomains. Again, note that we use Newton's method for the linearization due to its fast convergence in the neighborhood of the solution. Therefore, the method is called Newton-Krylov-FETI-DP or, in short, NK-FETI-DP. Let us note that this strategy is not restricted to the use of FETI-DP, but other domain decomposition approaches such as BDDC are possible to solve the resulting linearized system; see also Section 5.6.1. All together build the class of Newton-Krylov-Domain-Decomposition approaches.

As a starting point, we consider a nonlinear system of equations of the type $A(\hat{u}) = 0$, $\hat{u} \in V^h$, resulting from the equivalent minimization problem of a finite element discretization of a partial differential equation. Let us note that $A(\hat{u}) = 0$ operates on V^h, i.e., we consider finite element functions which are continuous in all interface variables. To use the same notation as before, we assume that we can reformulate the nonlinear problem $A(\hat{u}) = 0$ to

$$A(\hat{u}) = R^T K(R\hat{u}) - R^T f = 0. \tag{5.61}$$

Using N nonoverlapping subdomains Ω_i, $i = 1, \ldots, N$, to decompose the computational domain Ω, the quantities $K(u)$, f, and R can be represented as before as

$$K(u) = \begin{bmatrix} K_1(u_1) \\ \vdots \\ K_N(u_N) \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}, \quad R = \left[R_1^T, \ldots, R_N^T \right]^T,$$

where $K_i(u_i) - f_i$ represents the discretized nonlinear problem on subdomain Ω_i. Linearization of Equation (5.61) in the kth Newton iteration leads to the fully assembled linearized system

$$R^T DK(R\hat{u}^{(k)}) R \cdot \delta\hat{u}^{(k)} = R^T K(R\hat{u}^{(k)}) - R^T f \tag{5.62}$$

with the Newton iteration

$$\hat{u}^{(k+1)} = \hat{u}^{(k)} - \alpha^{(k)} \delta\hat{u}^{(k)}, \tag{5.63}$$

where $\alpha^{(k)}$ is a suitable step length. For example, $\alpha^{(k)}$ could be chosen such that the Wolfe conditions are fulfilled; see [77, 107, 122]. Using the restriction $u_i = R_i \hat{u}^{(k)}$ of the fully assembled vector $\hat{u}^{(k)}$, the operator $DK(R\hat{u}^{(k)})$ writes

$$DK(R\hat{u}^{(k)}) = \begin{bmatrix} DK_1(u_1) & & \\ & \ddots & \\ & & DK_N(u_N) \end{bmatrix},$$

i.e., $DK(R\hat{u}^{(k)})$ has the tangential matrices of the corresponding subdomains on its diagonal block entries.

Using a FETI-DP method for the solution of the linearized problem, we only consider a subassembled system in the primal variables instead of the fully assembled system; see Section 5.2. Therefore, we also require the jump operator B, which enforces continuity on the remaining interface variables Δ. To describe the relation between the fully assembled variables $\hat{u}$ and the partially assembled variables $\tilde{u}$, we introduce scaled versions of the assembly operators R_Π^T and R^T, which are denoted by $R_{\Pi,D}^T$ and R_D^T, respectively. For both matrices, the scaling factors depend on the multiplicity of the corresponding variables. Each row in $R_{\Pi,D}^T$, which corresponds to a primal variable, is multiplied with the inverse of its multiplicity. Analogously, each row in R_D^T is scaled with the multiplicity of the corresponding variable. Hence, we can write $\tilde{u} := R_{\Pi,D}^T R\hat{u}$ for the relation between $\tilde{u}$ and $\hat{u}$ and after solving the linearized equations, the relation between

the updates writes $\delta\hat{u} := R_D^T R_\Pi \delta\tilde{u}$, i.e., the (sub-)assembled variables are first restricted to the local subdomains and subsequently (partially) reassembled.

The subassembled system writes

$$\begin{bmatrix} R_\Pi^T DK(R_\Pi \tilde{u}^{(k)}) R_\Pi & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} R_\Pi^T K(R_\Pi \tilde{u}^{(k)}) - R_\Pi^T f \\ 0 \end{bmatrix}. \tag{5.64}$$

Recalling the notations

$$\begin{aligned} \widetilde{K}(\tilde{u}) &= R_\Pi^T K(R_\Pi \tilde{u}), \\ D\widetilde{K}(\tilde{u}) &= R_\Pi^T DK(R_\Pi \tilde{u}) R_\Pi, \text{ and} \\ \tilde{f} &= R_\Pi^T f \end{aligned}$$

from Section 5.3, system (5.64) can be written in compressed form as

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} \\ 0 \end{bmatrix}. \tag{5.65}$$

Note that system (5.65) is equivalent to the compressed system (5.8) in the linear case. From the second line of Equation (5.65), we obtain $B\delta\tilde{u}^{(k)} = 0 \; \forall \; k$ and, therefore, all iterates are continuous across the interface if the initial value is continuous across the interface.

One step of block elimination in Equation (5.65) leads to

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ 0 & F_{\mathrm{NK}}(\tilde{u}^{(k)}) \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} \\ d_{\mathrm{NK}}(\tilde{u}^{(k)}) \end{bmatrix}, \tag{5.66}$$

with

$$\begin{aligned} F_{\mathrm{NK}}(\tilde{u}^{(k)}) &= B \left(D\widetilde{K}(\tilde{u}^{(k)}) \right)^{-1} B^T, \\ d_{\mathrm{NK}}(\tilde{u}^{(k)} &= B \left(D\widetilde{K}(\tilde{u}^{(k)}) \right)^{-1} \left(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} \right). \end{aligned}$$

Hence, analogously to the linear case (cf. Equation (5.7)), we have to solve the reduced system

$$F_{\mathrm{NK}}(\tilde{u}^{(k)})\lambda = d_{\mathrm{NK}}(\tilde{u}^{(k)}), \tag{5.67}$$

which can be done by a Krylov subspace method using a preconditioner M^{-1} such as the standard Dirichlet preconditioner defined in [41, 150]; see also Section 5.2. The algorithmic overview of the NK-FETI-DP approach is given in Figure 5.5.

Init: $\tilde{u}^{(0)} \in \widehat{W}$
for $k = 1, \ldots,$ convergence **do**
  Build: $\widetilde{K}(\tilde{u}^{(k)})$ and $D\widetilde{K}(\tilde{u}^{(k)})$
  Solve (with Krylov subspace method):
  $B\left(D\widetilde{K}(\tilde{u}^{(k)})\right)^{-1} B^T \cdot \lambda = B\left(D\widetilde{K}(\tilde{u}^{(k)})\right)^{-1}\left(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f}\right)$ //see Equation (5.67)
  Compute $\delta\tilde{u}^{(k)}$ from λ:
  $\delta\tilde{u}^{(k)} = \left(D\widetilde{K}(\tilde{u}^{(k)})\right)^{-1}\left(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} + B^T\lambda\right)$ //see Equation (5.66)
  Update: $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\delta\tilde{u}^{(k)}$
end for

Figure 5.5: Newton-Krylov-FETI-DP algorithm. This algorithm has been similarly published in [107, Fig. 2.3] and [77].

Since $\widehat{W}$ is isomorphic to V^h, it is also possible to formulate a Newton iteration operating on $\widehat{W}$. With $u^{(k+1)}$, $u^{(k)}$, and $\delta u^{(k)} \in \widehat{W}$ and a suitable step length $\alpha^{(k)}$, we write

$$u^{(k+1)} = u^{(k)} - \alpha^{(k)}\delta u^{(k)},$$

where the Newton update $\delta u^{(k)} = R_\Pi \delta\tilde{u}^{(k)}$ results from the solution of Equation (5.65) by using $\tilde{u}^{(k)} := R_{\Pi,D}^T u^{(k)}$.

5.6 BDDC for Nonlinear Problems

The above presented framework for nonlinear FETI-DP methods (see Section 5.4) can be modified such that the nonlinear BDDC approach can be included. The nonlinear BDDC method was introduced in [77]. Before we introduce the nonlinear BDDC method into our framework (see Section 5.6.2), we shortly present the classical Newton-Krylov-BDDC approach (see Section 5.6.1), which is also used for the parallelization of the macroscopic problem in our FE2TI package when simulating the Nakajima test.

5.6.1 Newton-Krylov-BDDC

In this section, we briefly discuss the Newton-Krylov-BDDC approach based on the descriptions in [77].

As mentioned before, Newton-Krylov-BDDC belongs to the class of Newton-Krylov-Domain-Decomposition methods. The nonlinear problem is first linearized with Newton's method and afterwards decomposed into subdomains.

Let us reuse the previously introduced notation, i.e., the computational domain Ω is decomposed into N nonoverlapping subdomains Ω_i, $i = 1, \dots, N$. The discretized subdomain problems write $K_i(u_i) - f_i = 0$ and we have

$$K(u) := \begin{bmatrix} K_1(u_1) \\ \vdots \\ K_N(u_N) \end{bmatrix}, \quad f := \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}, \quad R = \left[R_1^T, \dots, R_N^T\right]^T, \quad u_i = R_i u;$$

see also Section 5.5. The NK-BDDC approach operates on the completely assembled system

$$R^T K(R\hat{u}) - R^T f = 0, \quad \hat{u} \in V^h. \tag{5.68}$$

Analogously to R, the restriction of the global interface variables to the local interface variables is denoted by R_Γ. Equation (5.68) is solved by Newton's method, which yields the iteration

$$\hat{u}^{(k+1)} = \hat{u}^{(k)} - \alpha^{(k)} \delta\hat{u}^{(k)},$$

where the update $\delta\hat{u}^{(k)}$ is computed from

$$R^T DK(R\hat{u}^{(k)}) R\delta\hat{u}^{(k)} = R^T K(R\hat{u}^{(k)}) - R^T f. \tag{5.69}$$

Accordingly to Equation (5.8), we partition the tangential matrix DK and the right-hand side into interior and interface variables I and Γ, respectively, which leads to

$$DK(R\hat{u}) = \begin{bmatrix} DK_{II}(R\hat{u}) & DK_{I\Gamma}(R\hat{u}) \\ DK_{\Gamma I}(R\hat{u}) & DK_{\Gamma\Gamma}(R\hat{u}) \end{bmatrix} \quad \text{and}$$

$$K(R\hat{u}) - f = \begin{bmatrix} (K(R\hat{u}) - f)_I \\ (K(R\hat{u}) - f)_\Gamma \end{bmatrix} = \begin{bmatrix} K_I(R\hat{u}) - f_I \\ K_\Gamma(R\hat{u}) - f_\Gamma \end{bmatrix};$$

see also Section 5.4.2.4. Now, to solve Equation (5.69) with the BDDC algorithm, we eliminate the interior variables and solve for the remaining variables, i.e., the assembled interface variables, by some preconditioned Krylov iteration. Eliminating the inner variables results in the Schur complement

$$S_{\Gamma\Gamma}(\hat{u}) = DK_{\Gamma\Gamma}(\hat{u}) - DK_{\Gamma I}(\hat{u}) DK_{II}(\hat{u})^{-1} DK_{I\Gamma}(\hat{u}).$$

As a preconditioner, we introduce

$$M^{-1}_{\mathrm{BDDC}}(\hat{u}) := R_D^T (R_\Pi^T DK(\hat{u}) R_\Pi)^{-1} R_D, \tag{5.70}$$

where the operator R_D^T is defined as

$$R_D^T := \begin{bmatrix} R_{\Delta,D}^T R_\Delta^B & 0 \\ 0 & I_\Pi \end{bmatrix}.$$

Here, $R_{\Delta,D}^T R_\Delta^B$ is the weighted restriction from the index set B to Δ. Since BDDC operates on the assembled system, the resulting preconditioned equation writes

$$M_{\mathrm{BDDC}}^{-1}(R\hat{u}^{(k)}) S_g(R\hat{u}^{(k)}) \delta\hat{u}_g^{(k)} = M_{\mathrm{BDDC}}^{-1}(R\hat{u}^{(k)}) g_g(R\hat{u}^{(k)}), \tag{5.71}$$

with

$$\begin{aligned} S_g(\hat{u}) &:= R_\Gamma^T S_{\Gamma\Gamma}(\hat{u}) R_\Gamma \quad \text{and} \\ g_g(\hat{u}) &:= R_\Gamma^T \left(K_\Gamma(\hat{u}) - f_\Gamma - DK_{\Gamma I}(\hat{u}) DK_{II}(\hat{u})^{-1} (K_I(\hat{u}) - f_I) \right). \end{aligned} \tag{5.72}$$

Finally, the Newton update $\delta\hat{u}^{(k)}$ is obtained as

$$\delta\hat{u}^{(k)} := \begin{bmatrix} \left(DK_{II}(R\hat{u}^{(k)})\right)^{-1} \left(K_I(R\hat{u}^{(k)}) - f_I - DK_{I\Gamma}(R\hat{u}^{(k)}) \delta\hat{u}_g^{(k)} \right) \\ \delta\hat{u}_g^{(k)} \end{bmatrix}.$$

5.6.2 Nonlinear BDDC Framework

With some further generalizations to the above introduced framework for nonlinear FETI-DP methods (see Section 5.4), it is possible to include the nonlinear BDDC approach (see [77]) into the framework. Similar to NL-4 (see Section 5.4.2.4), in a nonlinear BDDC method all interior variables are eliminated nonlinearly and the resulting Schur complement system on the interface variables Γ has to be linearized.

Based on the discussion in [85, Sec. 3], we describe how the nonlinear BDDC approach can be described in the context of a unified framework for nonlinear FETI-DP and BDDC methods.

So far, for nonlinear FETI-DP methods, we have considered nonlinearly right-preconditioned systems of the form $A(M(x)) = 0$ with $x = (\tilde{u}, \lambda)$ and A as defined in Equation (5.29); cf. Equation (5.25). Since BDDC methods operate on the assembled system, we do not need Lagrange multipliers to enforce continuity. Thus, we define $x := \hat{u}$. Accordingly, we define the nonlinear problem as

$$A(x) := A(\hat{u}) := R^T K(R\hat{u}) - R^T f. \tag{5.73}$$

Note that the same notation is used as before; see, e.g., Sections 5.5 and 5.6.1.

Similar to the NL-4 approach (see Section 5.4.2.4) as well as to the classical NK-BDDC approach (see Section 5.6.1), where the linearized system is partitioned into interior and interface variables, we now partition the nonlinear system into interior and interface variables. We obtain

$$A(\hat{u}) = \begin{bmatrix} A_I(\hat{u}) \\ A_\Gamma(\hat{u}) \end{bmatrix} = \begin{bmatrix} K_I(R\hat{u}) - f_I \\ R_\Gamma^T K_\Gamma(R\hat{u}) - R_\Gamma^T f_\Gamma \end{bmatrix}. \tag{5.74}$$

Analogously, we split the variables $\hat{u} = (\hat{u}_I, \hat{u}_\Gamma)$.

With a nonlinear right-preconditioner M, which fulfills the properties noted in Figure 5.1 (right), we can write the nonlinear BDDC method introduced in [77] in the context of our framework; see Figure 5.6. Applying Newton's method to $A(M(\hat{u})) = 0$ yields

$$DA(M(\hat{u}^{(k)})) \cdot DM(\hat{u}^{(k)})\delta\hat{u}^{(k)} = A(M(\hat{u}^{(k)}))$$

with the iteration

$$\hat{u}^{(k+1)} = \hat{u}^{(k)} - \alpha^{(k)}\delta\hat{u}^{(k)};$$

cf. Equations (5.26) and (5.27).

In the nonlinear BDDC approach, the interior variables are eliminated from the linearized system. Therefore, we introduce the nonlinear preconditioner

$$M(\hat{u}) := (M_I(\hat{u}), \hat{u}_\Gamma), \tag{5.75}$$

where $M_I(\hat{u})$ is the solution of

$$A_I(\hat{u}) = K_I(M_I(\hat{u}), R_\Gamma\hat{u}_\Gamma) - f_I = 0. \tag{5.76}$$

The latter equation is obtained from the first line of Equation (5.74) inserted into $A(M(\hat{u})) = 0$.

With the notation used in the context of nonlinear FETI-DP for building the tangents of A and M (see Section 5.4.1) we obtain

$$DA(M(\hat{u})) = \begin{bmatrix} DK_{II}(RM(\hat{u})) & DK_{I\Gamma}(RM(\hat{u}))R_\Gamma \\ R_\Gamma^T DK_{\Gamma I}(RM(\hat{u})) & R_\Gamma^T DK_{\Gamma\Gamma}(RM(\hat{u}))R_\Gamma \end{bmatrix}$$

from Equation (5.74). Furthermore, with the partial derivatives of Equation (5.76) and the definition of M in Equation (5.75), the derivative of M

Init: $x^{(0)}$
for $k = 0, \ldots,$ convergence **do**
  Compute: $g^{(k)} := M(x^{(k)})$ //see Figure 5.7 (left)
  if $||A(g^{(k)})||$ sufficiently small **then**
    break // Convergence of nonlinear right-preconditioned method
  end if

  Solve iteratively with some preconditioner:
  $DA(g^{(k)})DM(x^{(k)})\delta x^{(k)} = A(g^{(k)})$
  Update: $x^{(k+1)} = x^{(k)} - \alpha^{(k)}\delta x^{(k)}$
end for

Figure 5.6: Generalized nonlinearly algorithm. This figure has been similarly published in [85]; see [85, Fig. 5].

writes

$$DM(\hat{u}) = \begin{bmatrix} 0 & -DK_{II}^{-1}(RM(\hat{u}))DK_{I\Gamma}(RM(\hat{u}))R_\Gamma \\ 0 & I \end{bmatrix}.$$

From the implicit definition of the preconditioner M (see Equation (5.76)) and the system (5.74), we obtain

$$A(M(\hat{u})) = \begin{bmatrix} 0 \\ R_\Gamma^T K_\Gamma(RM(\hat{u})) - R_\Gamma^T f_\Gamma \end{bmatrix}.$$

Combining the last three equations, we obtain the nonlinear BDDC method as introduced in [77].

As in nonlinear FETI-DP, the product

$$DA(M(\hat{u})) \cdot DM(\hat{u}) = \begin{bmatrix} 0 & 0 \\ 0 & R_\Gamma^T DK_{\Gamma\Gamma}R_\Gamma - R_\Gamma^T DK_{\Gamma I}DK_{II}^{-1}DK_{I\Gamma}R_\Gamma \end{bmatrix} \quad (5.77)$$

yields the derivative of the nonlinear Schur complement. For simplicity, we have suppressed the function evaluation point $RM(\hat{u})$. Let us note that, again, we can remove the multiplication with the inner derivative $DM(\cdot)$ to obtain an additional update $\delta\hat{u}_I^{(k)}$ without changing the update $\delta\hat{u}_\Gamma^{(k)}$. As before, $\delta\hat{u}_I^{(k)}$ is only used to update the initial value for the computation of $g^{(k+1)}$, where

$$g^{(k)} = M(\hat{u}^{(k)}) = (M_I(\hat{u}^{(k)}), \hat{u}_\Gamma^{(k)})$$

is defined analogously to the nonlinear FETI-DP context.

For local convergence analysis of nonlinear BDDC methods, similar assumptions compared to nonlinear FETI-DP methods have to be made; see Assumptions 2 and 3 in Section 5.3. More details can be found in [85]. For a short comparison between nonlinear FETI-DP methods and nonlinear BDDC methods, we refer to [77].

The unified algorithmic overview including nonlinear FETI-DP and BDDC methods is presented in Figure 5.6.

5.7 Controlling the Inner Newton Iteration in Nonlinear Domain Decomposition

In the context of nonlinear FETI-DP and nonlinear BDDC methods, all variants except NL-1 have an inner and an outer Newton iteration. It is possible that all these methods show a loss of robustness and performance compared to the corresponding traditional Newton-Krylov-Domain-Decomposition variants, e.g., if the nonlinearities are not contained in the elimination set $\mathcal{E}$. In this case, the application of the preconditioner M may lead to an increase in the global energy. Furthermore, it is possible that the inner Newton iteration does not converge due to an insufficient coarse space. Therefore, we consider a strategy with some additional control of the inner Newton iteration to enlarge the convergence radius of our nonlinear FETI-DP methods. This approach was first introduced in [86] for nonlinear FETI-DP methods and was later generalized to the nonlinear BDDC method in [85]. The representations in this section are strongly based on [85, Sec. 4].

A common globalization strategy for Newton-type methods is the enforcement of a sufficient reduction of an energy, e.g., in each Newton iteration [146]. Therefore, we have to control the Newton update and a Newton step may have to be rejected and replaced, e.g., by a steepest descent step. Additionally, the step length usually has to be controlled to give some global convergence properties. This can be achieved by a line search approach satisfying, e.g., the Wolfe or Armijo condition; see [122]. For the use of a line search approach in nonlinear FETI-DP methods, we refer to [77]. Note that it is also possible to apply the previously introduced strategies to nonlinear right-preconditioned Newton-Krylov methods [86].

With the notation introduced in Equations (5.34) and (5.76), we can write the nonlinear elimination process performed by the preconditioner M as

$$A_{\mathcal{E}}(g^{(k)}) = 0,$$

Init: $g_0^{(k)} = x^{(k)}$, $l = 0$
while $||A_{\mathcal{E}}(g_l^{(k)})|| > \varepsilon_I$ **do**
  Update with Newton's method to $\boldsymbol{g_{l+1}^{(k)}}$
  $l = l + 1$
  $g^{(k)} = g_l^{(k)}$
end while

Init: $g_0^{(k)} = x^{(k)}$, $l = 0$, $\tau_{\text{tol}} \in (0, 1]$,
  $J_{\text{old}} = \frac{1}{2}||A(g_0^{(k)})||^2$
while $||A_{\mathcal{E}}(g_l^{(k)})|| > \varepsilon_I$ **do**
  Update with Newton's method to $\boldsymbol{g_{l+1}^{(k)}}$
  Compute: $J_{\text{new}} = \frac{1}{2}||A(g_{l+1}^{(k)}||^2$
  if $J_{\text{new}} > \tau_{\text{tol}} J_{\text{old}}$ **then**
    $g^{(k)} = g_l^{(k)}$
    break while
  else
    $J_{\text{old}} = J_{\text{new}}$
  end if
  $l = l + 1$
  $g^{(k)} = g_l^{(k)}$
end while

Figure 5.7: Computation of the preconditioner M with and without additional control of the inner Newton iteration. This figure has been published in [85]; see [85, Fig. 6].**Left:** Computation of $M(x^{(k)})$. **Right:** Computation of $\mathcal{M}(x^{(k)})$.

where $g^{(k)} = M(x^{(k)})$; see also Figure 5.7 (left). Performing the iteration described in Figure 5.7 (left), we minimize the energy $J_{\mathcal{E}} := \frac{1}{2}||A_{\mathcal{E}}(x)||^2$ without considering how the global energy $J = \frac{1}{2}||A(x)||^2$ evolves. In order to also obtain control over the global energy J, we check in each iteration how it develops. Our idea is to stop the inner Newton iteration whenever a sufficient descent in the global energy is not achieved. If we stop the inner iteration before convergence is reached, i.e., the nonlinear elimination is only performed approximately. The preconditioner representing the corresponding approximative nonlinear elimination is denoted with $\mathcal{M}(x^{(k)})$. If the sufficient descent condition $J(g_{l+1}^{(k)}) \leq \tau_{\text{tol}} J(g_l^{(k)})$ is not satisfied, we set $\mathcal{M}(x^{(k)}) := g_l^{(k)}$; see Figure 5.7 (right). Obviously, the parameter τ has to be chosen such that $0 < \tau_{\text{tol}} \leq 1$ is satisfied and throughout this thesis, we use $\tau_{\text{tol}} = 0.8$.

This strategy avoids oversolving in the inner Newton iteration and is somewhat similar to inexact Newton methods with carefully chosen forcing terms [36].

Note that we can end up with two extreme cases for $\mathcal{M}(\tilde{u}, \lambda)$. On one hand, we can end up with $\mathcal{M}(\tilde{u}, \lambda) = M(\tilde{u}, \lambda)$. On the other hand, even the first inner Newton iteration may not satisfy the descent condition. In this case, $\mathcal{M}(\tilde{u}, \lambda)$ reduces to the identity, which results in an NL-1 step in the context of nonlinear FETI-DP. Obviously, the number of inner Newton iterations within an outer

Newton iteration is never increased compared to the nonlinear methods without additional control over the global energy J in the inner Newton iteration.

Since we do not want to rely on the initial value $x^{(0)}$, we handle the very first computation of $\mathcal{M}(\tilde{u}, \lambda)$ in a slightly different way. Instead of stopping the inner Newton iteration if $J(g_1^{(0)}) > \tau_{\text{tol}} J(x^{(0)})$ is fulfilled, we continue computing $g_l^{(0)}$ until $J(g_l^{(0)}) > \tau_{\text{tol}} J(g_{l-1}^{(0)})$ holds for $l \geq 2$. This adoption can also be used for the computation of the initial value in NL-1; see Section 5.4.2.1. As a consequence of the usage of the approximation $\mathcal{M}(\tilde{u}, \lambda)$ instead of $M(\tilde{u}, \lambda)$, the product $DA \cdot D\mathcal{M}$ will be generally no longer identical to the derivative of the nonlinear Schur complement, i.e., Equations (5.45) and (5.77) do not hold anymore in the context of FETI-DP and BDDC, respectively. Furthermore, the block entries in the right-hand side corresponding to the elimination set $\mathcal{E}$ (see Equations (5.47) and (5.49)) can no longer be assumed to be zero since $\mathcal{M}$ is only an approximation of M. Therefore, in each outer Newton iteration, we solve the approximate tangential system

$$DA(\mathcal{M}(x^{(k)}))\delta x^{(k)} = A(\mathcal{M}(x^{(k)})).$$

Note that, as before, all linearized systems are identical at convergence.

In the following, we refer to nonlinear FETI-DP and BDDC methods using the approximation $\mathcal{M}$ instead of M as NL-ane-FETI-DP and NL-ane-BDDC, where the addition stands for approximate nonlinear elimination.

6 Numerical Results for Nonlinear FETI-DP Methods

In this chapter, we present results for our nonlinear FETI-DP methods described within this thesis for different model problems. On one hand, we consider different variants of the scaled p-Laplace equation with local and nonlocal nonlinearities. Local nonlinearities are completely contained within the subdomains while nonlocal nonlinearities cross the interface, i.e., the subdomain boundaries. On the other hand, we also consider a two-dimensional elasticity problem taking into account a Neo-Hooke material model. For the latter case, we only present sequential MATLAB [120] results.

First, we briefly introduce the model problems and the HPC systems that were used for the simulations; see Section 6.1 and Section 6.2. Afterwards, we give some general remarks on the implementation as well as on the different numbers that are presented in the tables; see Section 6.3. Finally, we present the numerical results; see Sections 6.4.1 to 6.4.3 and 6.5 to 6.8. Let us note that parts of this chapter were already published in this or similar form by the author of this thesis and his coauthors in [85, Sec. 5]. This includes most parallel results regarding the p-Laplace model problem.

6.1 Model Problems

6.1.1 The p-Laplace Equation

Let us first consider the p-Laplace equation. We start by defining the scaled p-Laplace operator for $p \geq 2$ as

$$\alpha\Delta_p u := \operatorname{div}(\alpha|\nabla u|^{p-2}\nabla u);$$

see also [107]. Subsequently, our model problem writes

$$\begin{aligned} -\alpha\Delta_p u - \beta\Delta_2 u &= 1 && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

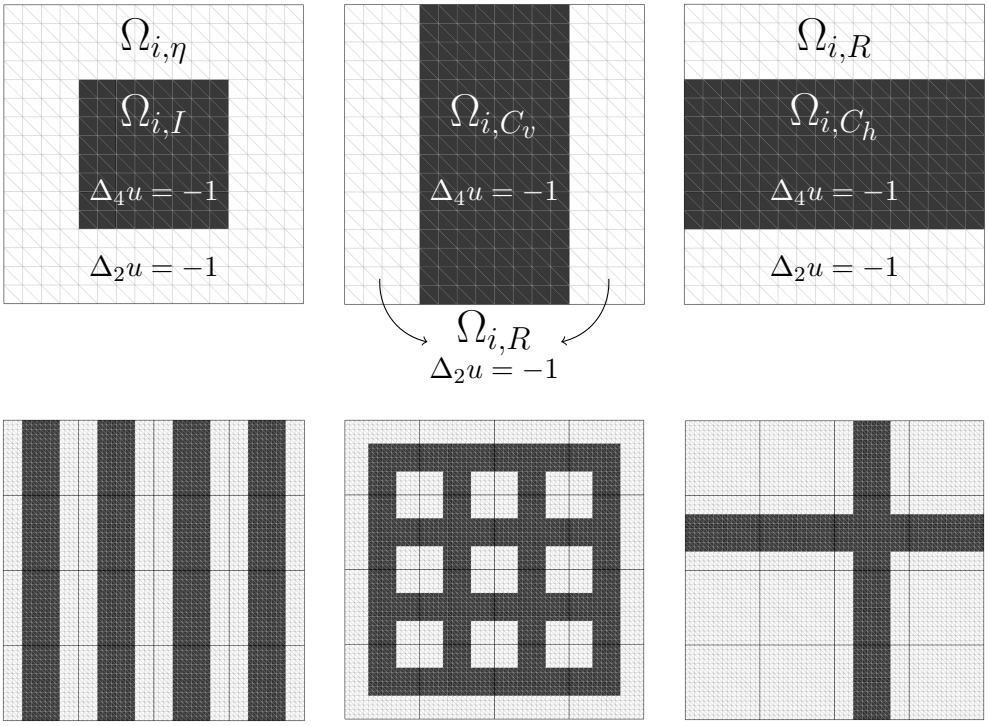


Figure 6.1: Top: Inclusion (left) as well as a vertical (middle) and horizontal (right) channel for a single subdomain Ω_i discretized using finite elements of diameter $h = 1/16$. Channels and inclusion have a width of $H/2$. **Bottom:** Different arrangements of horizontal and vertical channels. Vertical channels (left), "Grid" (middle), and "Cross" (right) for 16 subdomains discretized using finite elements with a typical diameter $h = 1/64$. All channels have a width $w_c = H/2$. This figure is similar to [85, Fig. 1].

where $\alpha, \beta : \Omega \to \mathbb{R}$ are coefficient functions. For further details regarding the p-Laplace equation as well as the introduction of the resulting energy functional, we refer to [107, Sec. 2.7.1]. Let us note that we exclusively use $p = 4$ throughout this thesis.

As mentioned before, we deal with different types on nonlinearities. On one hand we, consider problems where the nonlinearities are completely contained in the interior of the subdomains. On the other hand, we consider problems where we have nonlinearities on parts of the interface as well. The first case is called "**Localized Nonlinearities**" and the second case is called "**Nonlocal Nonlinearities**".

For "Localized Nonlinearities", we consider different types of inclusions. Let us note that we usually deal with square subdomains in two dimensions. There-

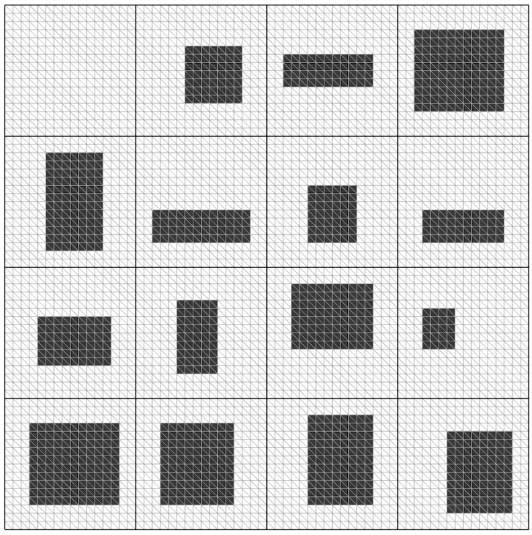


Figure 6.2: Example of random inclusions for 16 subdomains and finite elements with a diameter $h = 1/64$.

fore, the first choice of inclusions are squared inclusions of the same size for all subdomains, which is denoted *"Standard Inclusions"*; see Figure 6.1 (top left). In this case, the size of the inclusions can be characterized by the width η of the hull $\Omega_{i,\eta} = \Omega_i \setminus \Omega_{i,I}$. Usually η is defined in terms of the finite element diameter h. As a second case, we also think about other shapes of inclusions including a rectangle, a cross, and an ellipse (approximated on a regular grid). They are periodically arranged in x-direction; see [85, Fig. 8 (bottom)]. This case is referred to as *"Nonstandard Inclusions"*. In addition, we also consider random inclusions. Here, for each subdomain, the distance between the inclusion and the subdomain boundary is randomly chosen within a given range for all four sides separately. Thus, all inclusions have a rectangular shape, but in an extreme case, there might be subdomains without inclusions; see Figure 6.2 for an example. The latter case is referred to as *"Random Inclusions"*. We also consider a three-dimensional problem, where we have cubes as subdomains. We exclusively consider standard inclusions, where we have a centered spherical inclusion of radius $r = 0.3 \cdot H$ (approximated on a regular grid). We denote these parts of the subdomain Ω_i that belong to the inclusions with $\Omega_{i,I}$. Thus, $\Omega_I := \bigcup_{i=1}^{N} \Omega_{i,I}$ includes all finite element nodes that belong to the inclusions.

For all problems considering inclusions, we have no jumps in the coefficient functions α and β, i.e., we have

$$\alpha(x) = \begin{cases} 1 & \text{if } x \in \Omega_I \\ 0 & \text{elsewhere} \end{cases} \qquad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_I \\ 1 & \text{elsewhere;} \end{cases}$$

For the case of "Nonlocal Nonlinearities" we exclusively study two-dimensional problems. First, we consider an arrangement in which a centered channel of width w_c passes through each column of subdomains; see Figure 6.1 (top middle and bottom left). This is referred to as *"Channels"*. Furthermore, we consider a grid of channels that do not touch the boundaries of the computational domain Ω; see Figure 6.1 (bottom middle). This problem is denoted *"Grid"*. In addition, we also deal with a single channel in vertical direction as well as a cross of a single vertical and horizontal channel; see Figure 6.1 (bottom right). They are referred to as *"Single Channel"* and *"Cross"*, respectively. In all arrangements, we deal with different combinations of channels. Therefore, all parts belonging to $p = 4$ are collected in $\Omega_C := \bigcup_{i=1}^{N} \Omega_{i,C}$, where $\Omega_{i,C}$ is the part of each subdomain Ω_i that belongs to the channel(s). In general, $\Omega_{i,C}$ is further divided into parts of horizontal and vertical channels Ω_{i,C_h} and $\Omega_{i,C,v}$, respectively, i.e., $\Omega_{i,C} = \Omega_{i,C,v} \bigcup (\Omega_{i,C_h} \setminus (\Omega_{i,C_h} \cap \Omega_{i,C,v}))$; see Figure 6.1.

In contrast to "Localized Nonlinearities", we are not restricted to identical coefficients anymore. To be more precise, we also take into account higher coefficients inside the channels. Therefore, our coefficient functions write

$$\alpha(x) = \begin{cases} \{1, 10^5\} & \text{if } x \in \Omega_C \\ 0 & \text{elsewhere} \end{cases} \qquad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_C \\ 1 & \text{elsewhere.} \end{cases}$$

For two-dimensional problems, we consider the unit square $\Omega = [0, 1] \times [0, 1]$, different rectangles, and a curved domain. The latter is exclusively used for the case of "Nonstandard Inclusions". For three-dimensional problems, we consider cuboid domains.

6.1.2 Neo-Hooke Elasticity

As a second model problem, we also consider nonlinear hyperelasticity. Therefore, we take into account a Neo-Hooke material model. We exclusively deal with two-dimensional problems on the unit square. We consider a deformable body that is deformed by contact with a rigid arc. We consider frictionless contact using a penalty formulation as introduced in Section 4.2.4. Analogously to the assumption regarding the Nakajima test, the rigid tool moves in upward direction, i.e., only finite element nodes with $\overline{p}_y = 0$ can be in contact. This model problem is exclusively introduced for an analyzation of the NL-Res approach.

Let us consider nonlinear hyperelasticity a bit more detailed. The following discussion is based on [107, Sec. 3.4.1].

Since we consider two-dimensional problems, the strain energy density function (see [63, 152]) writes

$$W(u) = \frac{\mu}{2}\left(\mathrm{tr}(F^T F) - 2\right) - \mu \ln(J) + \frac{\lambda}{2}\ln^2(J);$$

see [107, Sec 3.4.1]. Here, we have $J = \det(F)$, $F(x) = \nabla\varphi(x)$ is, as before, the deformation gradient, and $\varphi(x) = x + u(x)$ is the deformation, where $u(x)$ denotes the displacement. Furthermore, λ and μ are the Lamé parameters, which can also be written in terms of Young's modulus E and the Poisson ratio ν as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)};$$

see [14, Sec. 6] and [131, Sec. 4]. Throughout this thesis, we consider $\nu = 0.3$, i.e., we deal with compressible hyperelasticity.

We are interested in the solution of the partial differential equation

$$\begin{aligned} -\mathrm{div}(P(F)) &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Omega_D, \end{aligned}$$

where $P(F)$ is the first Piola-Kirchhoff stress tensor, which is

$$P(F) = \mu(F - F^{-T}) + \lambda \ln(J) F^{-T};$$

see [107, Sec. 5.1].

Fore a more general form of the resulting energy functional functional, we refer to [107, Sec. 5.1]

6.2 Computational Platforms

Our simulations are performed on three different HPC systems in Germany, which belong to different classes of the German High Performance Computing Pyramid. They are listed in the following.

- JUQUEEN (Tier-1/0): 458 752 Blue Gene/Q cores (PowerPC A2 1.6 GHz; 16 cores and 16 GB per node); 5.9 PFlops; operated by Jülich Supercomputing Center (JSC) providing computing time for Germany and Europe; ranked 19th in the current TOP500 list (November, 2016).
- Taurus (Tier-2): 34 656 Xeon cores (2 020 nodes); 1.4 PFlops; operated by Center for Information Services and High Performance Computing (ZIH)

of the TU Dresden providing HPC resources for Saxony; TOP500 rank 107 (November, 2016).

- MagnitUDE (Tier-3): 13 536 cores (Broadwell XEON E5-2650v4 12C 2.2GHz; 24 cores and 72 GB per node); 476.5 TFlops NEC Cluster; operated by Center for Computational Sciences and Simulation (CCSS) of the Universität Duisburg-Essen (UDE) providing computing resources for UDE; TOP500 rank 384 (November, 2016).

On Taurus, we use the Haswell XEON E5-2680v3 12C 2.5GHz processor partition with 24 cores and 64 GB memory per node.

6.3 General Remarks

Before we give some implementation remarks, we first discuss the notation in the following sections. In all our tables, we refer to the traditional NK-FETI-DP approach as NK. As already introduced before, our nonlinear FETI-DP variants are referred to as NL-Res as well as NL-X, $X = 1, 2, 3, 4$ and NL-ane-X, respectively. Note that NL-ane-1 is not considered in this thesis. In all our simulations, we exclusively deal with linear finite elements (P_1) and only subdomain vertices on the interface are chosen as primal variables. For the effect of different coarse spaces enforced by the transformation of basis approach, we refer to [77].

To provide a fair comparison in terms of runtime for our parallel implementation of our nonlinear FETI-DP variants as well as the traditional NK-FETI-DP approach, we use a common software framework. Thus, we use the same software building blocks. The software is implemented in PETSc [4–6] and we use the latest version of UMFPACK [28] for all local sparse factorizations. Note that we also present some results obtained from our sequential MATLAB [120] implementation for which no runtimes are shown. In order to interpret the parallel as well as the sequential results in a similar way, we provide the number of necessary factorizations of the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$ (denoted by "Coarse Fact."), the number of local factorizations of $DK_{\mathcal{E}\mathcal{E}}$ (denoted by "Local Fact.") and the accumulated sum of Krylov iterations over all outer Newton iterations (denoted by "Krylov Its."). This is identical to previous publications; see [77, 80, 85].

For NL-1 and NL-2, the factorization of the coarse problem is not only necessary in the outer Newton iterations, but also in the computation of the initial value for NL-1 as well as for the inner loop for NL-2. Therefore, we consider the

number of necessary coarse factorizations of $\widetilde{S}_{\Pi\Pi}$ in the inner and outer loop separately. Coarse factorizations in the inner and outer loops are denoted by "in." and "out.", respectively. The lowest numbers of the considered quantities are marked in bold. Let us note that the number of inner coarse factorizations is always zero for NL-3 and NL-4 and for all methods, the number of outer coarse factorizations is identical to the number of outer Newton iterations.

For our recent MATLAB implementation of NL-Res, we also present the average size of the elimination set $\mathcal{E}$, since its size depends on the chosen parameters δ and ρ_{res}.

For all parallel results, we focus on the overall execution time (denoted by "exec. time"), which includes the time to assemble and to solve the problem. As before, the lowest runtimes are also marked in bold. Furthermore, the number of subdomains is always equal to the number of MPI ranks. As a baseline for the computation of the parallel efficiency, we always choose the fastest approach on the smallest number of compute cores. Thus, we obtain parallel efficiencies below 100 % for four of five approaches for the smallest computations. Note that we only have five different approaches, since we do not have a parallel implementation of NL-Res yet.

A fair comparison of our nonlinear FETI-DP algorithms also require the usage of identical stopping criteria for the inner as well as for the outer Newton iteration. We consider stopping criteria that are formulated in terms of the variable $\tilde{u}$. The outer Newton iteration is usually stopped whenever the fully assembled nonlinear residual is smaller than a pre-chosen tolerance ε_O. For "Nonstandard Inclusions" on the curved domain as well as for three-dimensional problems, we formulate stopping criteria based on the norm of the update $\delta\tilde{u}$; see the caption of the tables. The inner Newton iteration is associated with the solution of $\widetilde{K}_{\mathcal{E}}(M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}},\lambda),\tilde{u}_{\mathcal{L}}) + B_{\mathcal{E}}^T\lambda - \tilde{f}_{\mathcal{E}} = 0$; see Equation (5.34). Similar to the outer Newton iteration, we also have a pre-chosen tolerance ε_I corresponding to the inner Newton iteration. To avoid unnecessary exactness in the first outer Newton iterations without loosing sufficient exactness at convergence, inner Newton iterations are stopped if $||\widetilde{K}_{\mathcal{E}}(M_{\tilde{u}_{\mathcal{E}}}(\tilde{u}_{\mathcal{L}},\lambda),\tilde{u}_{\mathcal{L}}) + B_{\mathcal{E}}^T\lambda - \tilde{f}_{\mathcal{E}}||_{L_2}$ is smaller than the minimum of ε_I and the norm of the fully assembled residual of the previous outer Newton iteration multiplied by 10^{-2}. Throughout our simulations we deal with different values $\varepsilon_O \in \{10^{-8}, 10^{-12}\}$ and $\varepsilon_I \in \{10^{-5}, 10^{-6}, 10^{-7}\}$; see the captions of the corresponding tables and figures.

It can be shown for the p-Laplace equation that the tangential matrix is always symmetric positive definite if it is not evaluated in constant functions. In our simulations, we always consider zero Dirichlet boundary conditions on

the complete boundary and we have a nonzero initial value. Thus, we usually use the preconditioned CG (PCG) method as a Krylov subspace method. If we consider inexact reduced FETI-DP variants, the block-triangular preconditioner is not symmetric. Instead of using the symmetric positive definite reformulation (cf. [15, 92]), we use GMRES as a Krylov subspace method. The latter is also chosen for the sequential MATLAB implementation of the nonlinear hyperelasticity problem with contact.

No matter which Krylov subspace method is considered, we use a relative residual tolerance of 10^{-10} for all Krylov iterations. For sure, this leads to an unnecessary high exactness, especially in the first outer Newton iterations. More advanced techniques to choose forcing terms in inexact Newton's method can be found in [36], but is not in the focus of this thesis.

6.4 Localized Nonlinearities in Two Dimensions

In this section we analyze the different nonlinear FETI-DP variants as well as the traditional NK-FETI-DP approach for nonlinearities that are completely contained within the subdomains. We split the results into two parts. On one hand, we consider the results on midsized HPC systems, namely magnitUDE and Taurus (see Section 6.2), where we exclusively use the exact FETI-DP approaches. On the other hand, we also consider numerical results on JUQUEEN, which was one of the largest HPC systems of their time. For JUQUEEN, we can scale to more than 100k MPI ranks and, therefore, we also take into account the inexact reduced versions of our FETI-DP methods.

6.4.1 Standard, Exact Nonlinear FETI-DP Methods

Let us first consider our standard FETI-DP methods for nonlinear problems, i.e., with solving the coarse problem with a sparse direct solver. Besides the different NL-X variants, $X = 1, \dots, 4$, we also consider the traditional NK-FETI-DP approach. At first, we discuss the results obtained from simulations on Taurus and magniUDE. These HPC systems have a smaller number of compute cores than the JUQUEEN [69], but provide more memory per core. Thus, we can deal with comparably large subdomains and we choose $H/h = 400$, which results in 160k d.o.f. per subdomain. Let us recall that H and h are the maximum diameters of the subdomains and finite elements, respectively.

Since the magnitUDE has only 13 536 cores, we have to use two MPI ranks per core (making use of the hyperthreads for MPI processes) for larger problems. As a consequence, we also choose two MPI ranks per core for smaller problems.

For our applications, we have observed a small performance gain of about 10 % if we use two MPI ranks per core instead of one. Let us note that we do not use threading.

On Taurus, we always use one MPI rank per core, i.e., the number of MPI ranks is identical to the number of cores and to the number of subdomains.

In this section, we exclusively deal with "Localized Nonlinearities", i.e., the nonlinearities are completely contained in the elimination sets of NL-2, NL-3, and NL-4. As a consequence, the inner Newton iterations of these methods are expected to be effective in reducing the number of outer Newton iterations. Of course, NL-1 only has an outer Newton iteration, but the computation of the initial value (see Equation (5.53)) is somehow equivalent to an inner iteration of NL-2 and is therefore also expected to be effective.

For "Standard Inclusions", we present results for simulations on Taurus as well as on magnitUDE; see Figure 6.3 and Table 6.1. For more details regarding the simulation on Taurus, we refer to [85, Tab. 2], where all simulation results are presented in a table analogously to the tables in this thesis. On both HPC systems, we use a rectangular domain $\Omega = [0, 1.5] \times [0, 1]$, but the size of the inclusions as well as the tolerance for the inner Newton iteration slightly differ; see the captions of Figure 6.3 and Table 6.1.

For "Nonstandard Inclusions", we present results for a simulation on magnitUDE with a curved domain Ω; see Table 6.2.

The simulation results presented in this section have in common that NL-2 is the fastest method and is therefore especially faster than NK; see Figure 6.3 and Tables 6.1 and 6.2. For a small number of MPI ranks, NL-4 is always slower than NK, but it turns out for "Standard Inclusions" that NK is always the slowest method beyond 96 ranks; see Figure 6.3 and Table 6.1. For "Nonstandard Inclusions", we obtain that NK and NL-4 have identical computing times for the largest test; see Table 6.2.

It turns out that the nonlinear FETI-DP algorithms NL-2, NL-3, and NL-4 are about twice as fast compared to NK for large problems of the type "Standard Inclusions". All nonlinear FETI-DP variants drastically reduce the number of Krylov iterations compared to the traditional NK approach. This is achieved by investigating more local work; see the number of local factorizations. Again, let us note that we refer to [85, Tab. 2] for a detailed presentation of the simulation results on Taurus.

Following the discussion in Section 5.4.3, the NL-2 approach has the most expensive inner Newton iteration since each inner iteration requires a factorization of the coarse problem. However, this also means that each inner iteration

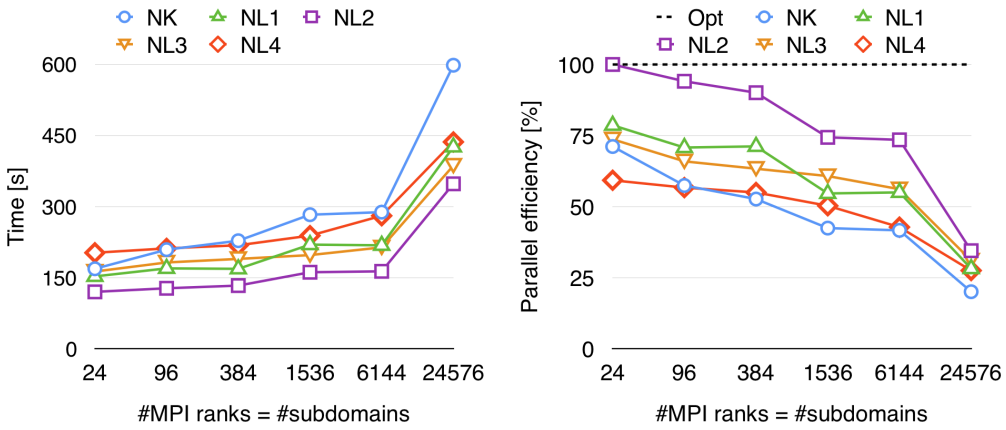


Figure 6.3: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); weak scalability of nonlinear FETI-DP algorithms (NL-X, $X = 1, \ldots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); domain $\Omega = [0, 1.5] \times [0, 1]$ decomposed into square subdomains; $p = 4$ and a weight of $\alpha = 1$ inside the inclusions and $p = 2$ and $\beta = 1$ elsewhere; $H/h = 400$; $\eta = 10h$; $\varepsilon_I = 10^{-6}$; $\varepsilon_O = 10^{-12}$; one MPI rank per core; computed on Taurus. For further information; see [85, Tab. 2].

provides a global transport of information. In contrast, in NL-3 and NL-4, we have completely local nonlinear problems, i.e., the inner iteration does not require any factorization of the coarse problem. As a consequence, there is no global transport of information. We obtain that the NL-2 method is most effective since it achieves the largest reduction of Krylov iterations and the fastest computing times. As a result, the savings in the accumulated number of coarse solves (see "Coarse Fact.") cannot compensate for the higher numbers of local factorizations and Krylov iterations. Thus, the cost of the coarse problems does not seem to be significant enough for the considered problem sizes.

It is not surprising that the computing times of NL-1 are always in between the computing times of NK and the remaining nonlinear FETI-DP methods. This can be explained by the fact that we have no inner iteration in NL-1 and, therefore, NL-1 is algorithmically closely related to the traditional NK approach.

It is striking that the parallel efficiencies significantly drop for the largest computations for "Standard Inclusions" on both HPC systems due to the costs resulting from solving the coarse problem exactly. For example, the parallel efficiency for NL-2 decreases from above 70 % to below 50 %; see Figure 6.3

Table 6.1: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); nonlinear FETI-DP algorithms (NL-X, $X = 1, \dots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); domain $\Omega = [0, 1.5] \times [0, 1]$ decomposed into square subdomains; $p = 4$ and a weight of $\alpha = 1$ inside the inclusions and $p = 2$ and $\beta = 1$ elsewhere; $H/h = 400$; $\eta = 20h$; $\varepsilon_I = 10^{-7}$; $\varepsilon_O = 10^{-12}$; two MPI ranks per core; computed on magnitUDE. This table has already been published in [85]; see [85, Tab. 1].

Localized Nonlinearities - Standard Inclusions **2D; $p = 4$; $H/h = 400$; exact FETI-DP; computed on magnitUDE**								
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**Coarse Fact.**		**Krylov Its.**	**Exec. Time**	**Parallel Effic.**
				in.	**out.**			
24	3 844 001	NK	**20**	-	20	363	171.01s	63%
		NL-1	23	11	12	224	142.64s	76%
		NL-2	26	22	**4**	**73**	**108.09s**	**100%**
		NL-3	40	**0**	5	91	148.77s	73%
		NL-4	42	**0**	5	98	171.81s	63%
96	15 368 001	NK	**19**	-	19	499	191.08s	57%
		NL-1	25	12	13	345	167.56s	65%
		NL-2	27	23	**4**	**105**	**119.05s**	**91%**
		NL-3	43	**0**	5	132	166.60s	65%
		NL-4	37	**0**	5	144	164.66s	66%
384	61 456 001	NK	**21**	-	21	619	222.28s	49%
		NL-1	25	12	13	351	176.12s	61%
		NL-2	29	25	**4**	**117**	**130.29s**	**83%**
		NL-3	43	**0**	5	144	173.42s	62%
		NL-4	38	**0**	5	162	176.77s	61%
1 536	245 792 001	NK	**24**	-	24	738	265.48s	41%
		NL-1	33	12	21	541	250.05s	43%
		NL-2	30	26	**4**	**120**	**136.43s**	**79%**
		NL-3	43	**0**	5	150	175.94s	61%
		NL-4	41	**0**	5	168	190.05s	57%
6 144	983 104 001	NK	**25**	-	25	802	297.77s	36%
		NL-1	29	15	14	411	219.46s	49%
		NL-2	32	28	**4**	**125**	**149.87s**	**72%**
		NL-3	47	**0**	5	157	196.45s	55%
		NL-4	45	**0**	5	173	213.16s	51%
24 576	3 932 288 001	NK	**26**	-	26	871	485.9s	22%
		NL-1	29	15	14	400	313.19s	35%
		NL-2	35	31	**4**	**127**	**225.31s**	**48%**
		NL-3	47	**0**	5	159	235.17s	46%
		NL-4	43	**0**	5	177	240.28s	45%

and Table 6.1. As already discussed in Section 5.4.4, we can improve scalability, especially for a large number of cores and ranks, if we use inexact reduced versions of our FETI-DP methods. Therefore, we also consider our FETI-DP algorithms in combination with a multilevel solver for the coarse problem in the next section.

Table 6.2: Model problem "Localized Nonlinearities" - "Nonstandard Inclusions" (see Section 6.1.1); nonlinear FETI-DP algorithms (NL-X, $X = 1, \dots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); $p = 4$ and a weight of $\alpha = 1$ inside the inclusions and $p = 2$ and $\beta = 1$ elsewhere; domain Ω is a curved domain with a height of 0.1 and a width of 1.0, see also [85, Fig. 8 (bottom)]; decomposed into square subdomains; $H/h = 400$; $\varepsilon_I = 10^{-5}$; $\varepsilon_O = 10^{-8}$; the stopping criterion is based on the norm of $\delta\tilde{u}$; two MPI ranks per core; computed on magnitUDE. This table has already been published in [85]; see [85, Tab. 3].

Localized Nonlinearities - Nonstandard Inclusions **2D; $p = 4$; $H/h = 400$; exact FETI-DP; computed on magnitUDE**								
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**Coarse Fact.**		**Krylov Its.**	**Exec. Time**	**Parallel Effic.**
				in.	**out.**			
24	3 844 001	NK	**19**	-	19	343	88.27s	60%
		NL-1	20	11	9	138	60.92s	87%
		NL-2	23	19	**4**	**62**	**53.20s**	**100%**
		NL-3	40	**0**	6	92	84.67s	63%
		NL-4	54	**0**	8	128	125.39s	42%
96	15 368 001	NK	**21**	-	21	568	116.85s	46%
		NL-1	27	9	18	350	107.51s	49%
		NL-2	31	26	**5**	**107**	**75.58s**	**70%**
		NL-3	41	**0**	6	142	91.81s	58%
		NL-4	50	**0**	9	229	135.84s	39%
384	61 456 001	NK	**22**	-	22	614	125.49s	42%
		NL-1	26	10	16	332	101.88s	52%
		NL-2	27	23	**4**	**95**	**67.11s**	**79%**
		NL-3	33	**0**	6	150	79.95s	67%
		NL-4	44	**0**	9	243	127.28s	41%
1536	245 792 001	NK	**25**	-	25	729	152.78s	35%
		NL-1	27	8	19	380	116.05s	46%
		NL-2	32	27	**5**	**111**	**81.75s**	**65%**
		NL-3	37	**0**	6	155	89.88s	59%
		NL-4	43	**0**	8	246	126.53s	42%
6144	983 104 001	NK	**20**	-	20	610	136.54s	39%
		NL-1	28	8	20	378	127.45s	42%
		NL-2	29	25	**4**	**98**	**77.69s**	**68%**
		NL-3	36	**0**	6	157	92.28s	58%
		NL-4	45	**0**	8	252	136.83s	39%

6.4.2 Scalability on JUQUEEN

In this section, we present scaling results on the JUQUEEN supercomputer [69] for our exact FETI-DP methods as well as for inexact reduced FETI-DP methods. In contrast to the previous section, we consider smaller subdomains with H/h=200 due to the smaller amount of memory per core and the slower BlueGene/Q PowerPC cores compared to the other x86-based supercomputers.

Similar to the magnitUDE, we use two MPI ranks per core. It is not motivated by the simulation of larger problems, but by the efficient use of hardware threads of the BlueGene/Q processor. By using threading or multiple MPI ranks per core, we can achieve a significant performance gain of nearly a factor two; see also [79].

For exact FETI-DP algorithms, we consider the unit square $\Omega = [0,1] \times [0,1]$ (see Table 6.3 and Figure 6.4) as well as a rectangle $\Omega = [0,2] \times [0,1]$ (see Figure 6.5) as computational domains. As one can see in Figure 6.5, the execution times increase for a problem size of 32 768 MPI ranks. Simultaneously, the parallel efficiency drops due to the exact solution of the coarse problem. Therefore, we consider a maximum problem size of 16 384 MPI ranks for the unit square; see Table 6.3 and Figure 6.4. It turns out that the results in Table 6.3 and Figures 6.4 and 6.5 are similar to the results presented in the previous section; see Section 6.4.1.

We obtain good weak scalability for all our nonlinear FETI-DP methods for a maximum problem size of 16 384 MPI ranks even if we do not consider inexact reduced FETI-DP variants. This can be explained by the powerful network of BlueGene/Q machines. Since also the traditional NK approach suffers from the exact factorization of the coarse problem, all nonlinear FETI-DP methods are significantly faster compared to the traditional NK approach; see Table 6.3 and Figures 6.4 and 6.5.

As mentioned before, parallel scalability can be improved if we do not solve the coarse problem exactly. Thus, we also consider inexact reduced FETI-DP algorithms, where we apply an AMG preconditioner to the coarse problem instead of using an exact (sparse) direct solver; see Section 5.4.4. The simulations have been performed on the computational domain $\Omega = [0,2] \times [0,1]$ and the results are presented in Table 6.4 and Figure 6.6. We obtain weak parallel scalability to 131 072 MPI ranks. To save some space, we have only shown the smallest as well as the largest problem sizes in Table 6.4. For the iteration numbers regarding the problem sizes in between, we refer to [85, Tab. 5].

As a consequence of the application of the AMG preconditioner, we have to perform additional Krylov iterations for the solution of

$$M_{\mathrm{AMG}}^{-1} \widetilde{S}_{\Pi\Pi} \delta\tilde{u}_{\Pi} = M_{\mathrm{AMG}}^{-1} \cdot \mathrm{rhs}_{\Pi}$$

in the inner loop of NL-2 as well as in the computation of the initial value of NL-1; see also [80] for further details. Since these Krylov iterations are cheaper compared to those in the outer Newton iteration, we count them separately and present them in an extra sub-column in Table 6.4 that is denoted "$\widetilde{S}_{\Pi\Pi}$". The

Table 6.3: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); nonlinear FETI-DP algorithms (NL-X, $X = 1, \dots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); $p = 4$ and a weight of $\alpha = 1$ inside the inclusions and $p = 2$ and $\beta = 1$ elsewhere; domain $\Omega = [0,1] \times [0,1]$ and $H/h = 200$ decomposed into square subdomains; $\eta = 10h$; $\varepsilon_I = 10^{-7}$; $\varepsilon_O = 10^{-12}$; two MPI ranks per core; computed on JUQUEEN [69]; see also Figure 6.4. This table has already been published in [85]; see [85, Tab. 4].

Localized Nonlinearities - Standard Inclusions **2D; $p = 4$; $H/h = 200$; exact FETI-DP; computed on JUQUEEN**								
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**Coarse Fact.**		**Krylov Its.**	**Exec. Time**	**Parallel Effic.**
				in.	**out.**			
64	2 563 201	NK	**21**	-	21	443	236.96s	56%
		NL-1	**21**	15	6	126	144.82s	91%
		NL-2	23	20	**3**	**66**	**131.68s**	**100%**
		NL-3	36	**0**	5	105	193.75s	68%
		NL-4	38	**0**	6	135	227.14s	58%
256	10 246 401	NK	**22**	-	22	559	261.73s	50%
		NL-1	**22**	15	7	180	160.37s	82%
		NL-2	23	20	**3**	**79**	**133.47s**	**99%**
		NL-3	37	**0**	5	127	201.16s	66%
		NL-4	39	**0**	6	166	237.00s	56%
1024	40 972 801	NK	**24**	-	24	660	294.08s	45%
		NL-1	26	16	10	241	200.32s	66%
		NL-2	29	25	**4**	**103**	**171.35s**	**77%**
		NL-3	38	**0**	5	134	207.26s	64%
		NL-4	39	**0**	6	177	239.97s	55%
4096	163 865 601	NK	**26**	-	26	770	336.14s	39%
		NL-1	**26**	16	10	248	209.08s	63%
		NL-2	29	25	**4**	**107**	**181.57s**	**73%**
		NL-3	39	**0**	5	139	215.60s	61%
		NL-4	38	**0**	6	185	239.30s	55%
16384	655 411 201	NK	**27**	-	27	823	403.87s	33%
		NL-1	28	19	9	216	250.36s	53%
		NL-2	31	27	**4**	**110**	**230.21s**	**57%**
		NL-3	41	**0**	5	141	239.12s	55%
		NL-4	41	**0**	6	188	268.49s	49%

Krylov iterations that are required for the outer Newton iteration are referred to as "Full" in Table 6.4. Furthermore, the application of the AMG preconditioner replaces a coarse factorization. Therefore, we count the different AMG setups instead of coarse factorizations in Table 6.4. Due to the large number of MPI ranks, we usually only use a small fraction of the available MPI ranks for solving the coarse problem. For example, we only use 2 % for the largest computation on 131 072 MPI ranks; see also [80].

The results obtained from the simulations using inexact reduced FETI-DP methods are qualitatively similar to the results obtained from simulations using exact FETI-DP algorithms; see Section 6.4.1 and Table 6.3. For the first time,

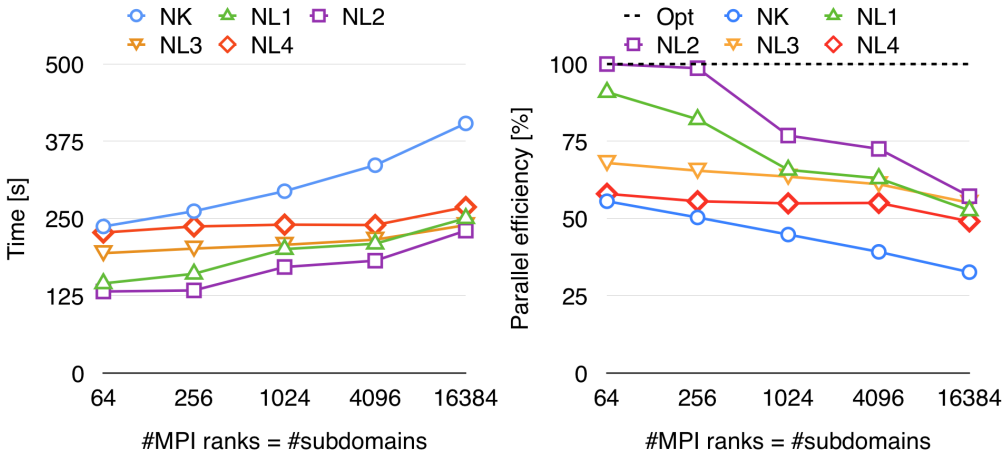


Figure 6.4: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); weak scalability of nonlinear FETI-DP algorithms (NL-X, $X = 1, \ldots, 4$) and the more traditional Newton-Krylov-FETI-DP method (NK) on the JUQUEEN BlueGene/Q supercomputer at Forschungszentrum Jülich [69]; data from Table 6.3. This figure has partly been published in [85]; see [85, Fig. 9].

the NL-3 approach is the fastest method for and beyond 8 192 MPI ranks. We benefit from the favorable inner Newton iteration without solving any coarse problem as well as the small number of outer Newton iterations, which additionally leads to a small number of Krylov iterations. Even if the NL-4 approach is very similar to the NL-3 approach, it is slower due to a larger number of Krylov iterations.

6.4.3 Localized Nonlinearities in Three Dimensions

We now consider "Localized Nonlinearities" in three dimensions. As mentioned in Section 6.1.1, we exclusively deal with centered spherical inclusions (approximated on a regular grid) of the same size for each subdomain. We always use a radius of $r = 0.3H$ for the inclusions. Let us note that we have chosen the stopping criterion based on the norm of the update $\delta\tilde{u}$ instead of the norm of the nonlinear residual for all simulations presented within this section.

All in all, we get similar results as for two dimensions. At the cost of more local work, the numbers of Krylov iterations and outer Newton iterations for NL-X, $X = 1, \ldots, 4$, are significantly reduced compared to NK. However, the costs of the coarse problem in 3D seem to be relevant even for smaller problem

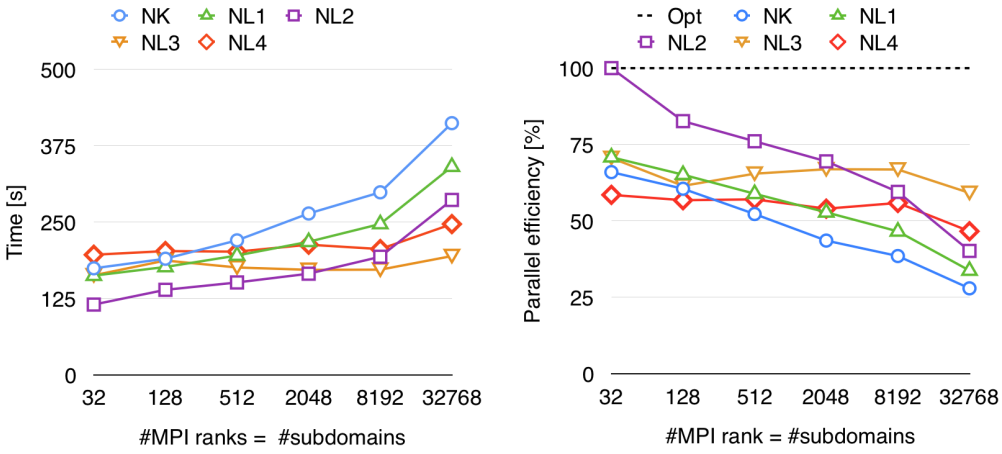


Figure 6.5: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); weak scalability of nonlinear FETI-DP algorithms (NL-X, $X = 1, \ldots, 4$) and the more traditional Newton-Krylov-FETI-DP method (NK) on the JUQUEEN BlueGene/Q supercomputer at Forschungszentrum Jülich [69]; same settings as in Figure 6.4 except for the computational domain $\Omega = [0, 2] \times [0, 1]$ and $\varepsilon_I = 10^{-6}$.

sizes, since one of the methods NL-3 and NL-4 is the fastest for each of the two computational domains Ω; see Tables 6.5 and 6.6.

Let us first consider the smaller domain $\Omega = [0, 1.5] \times [0, 1] \times [0, 1]$; see Table 6.5. While NL-2 and NL-3 have almost identical execution times, the NL-4 approach achieves the fastest convergence except for the smallest test. This is related to a reduced number of Krylov iterations and a much lower number of coarse solves in comparison to NL-2. At the same time, the local work increases only slightly. The better performance compared to NL-3 is partly due to the fact that the average timer per inner iteration is significantly lower; see Figure 6.7.

If we consider $\Omega = [0, 4] \times [0, 4] \times [0, 3]$, we get slightly different results; see Table 6.6. Instead of NL-4, NL-3 is the fastest method. The inner iteration of NL-4 is not as effective as before, which leads to an increased number of outer Newton iterations and, thus, to more Krylov iterations. This could be related to the larger interface. Nevertheless, the NL-4 algorithm is considerably faster than the traditional NK approach for the largest test. The shorter execution time of NL-3 compared to NL-1 and NL-2 results from a similar amount of local work. We benefit from the fact that the local factorization for NL-3 does not include the coarse problem.

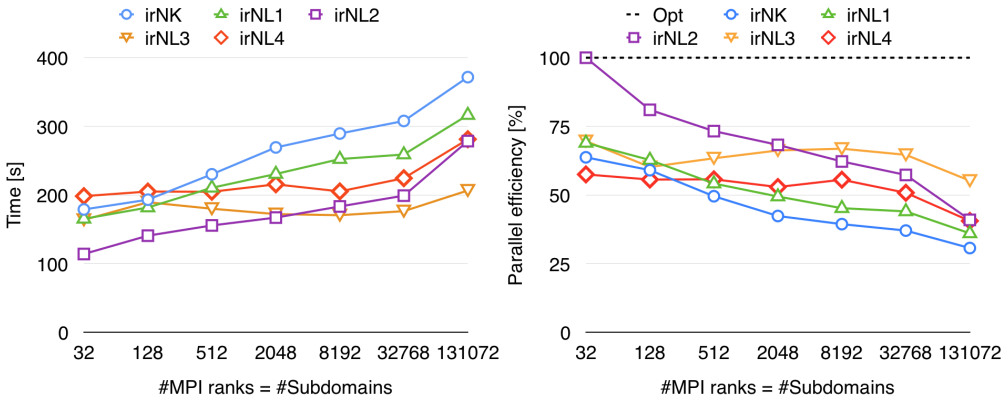


Figure 6.6: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); weak scalability of new inexact reduced FETI-DP algorithms (irNL-X, $X = 1, \ldots, 4$) and the inexact reduced version of the more traditional Newton-Krylov-FETI-DP algorithm (irNK) on the JUQUEEN BlueGene/Q machine at Forschungszentrum Jülich [69]; data from Table 6.4. This figure has partly been published in [85]; see [85, Fig. 10].

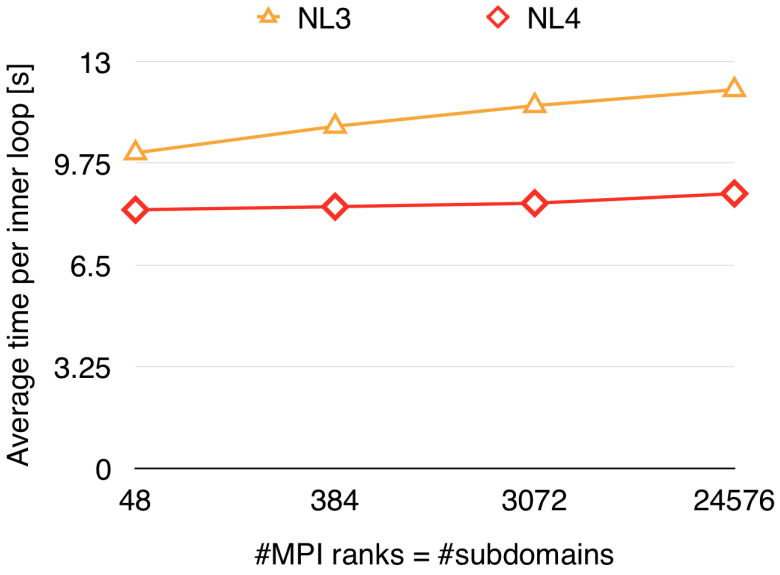


Figure 6.7: Model problem "Localized Nonlinearities" in 3D (see Section 6.1.1); comparison of the weak scalability behavior of the inner loops of NL-3 and NL-4. Here, we present the average runtime per Newton step for computations performed on magnitUDE; see Table 6.6 for the complete results. This figure has already been published in [85]; see [85, Fig. 11].

Table 6.4: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); inexact reduced nonlinear FETI-DP algorithms (irNL-X, $X = 1, \dots, 4$) compared to the inexact reduced version of the more traditional Newton-Krylov-FETI-DP method (NK); domain $\Omega = [0, 2] \times [0, 1]$ decomposed into square subdomains; $p = 4$ and a weight of $\alpha = 1$ inside the inclusions and $p = 2$ and $\beta = 1$ elsewhere; $H/h = 200$; $\eta = 10h$; $\varepsilon_I = 10^{-6}$; $\varepsilon_O = 10^{-12}$; two MPI ranks per core. Instead of the exact factorizations of $\widetilde{S}_{\Pi\Pi}$, we now have to set up an AMG preconditioner for $\widetilde{S}_{\Pi\Pi}$ several times. We also have one AMG application per GMRES iteration; computed on the JUQUEEN supercomputer [69]; see also Figure 6.6. This table has similarly been published in [85]; see [85, Tab. 5].

Localized Nonlinearities - Standard Inclusions **2D; $p = 4$; $H/h = 200$; inexact reduced FETI-DP; computed on JUQUEEN**									
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**AMG Setup**		**Krylov Its.**		**Exec. Time**	**Parallel Effic.**
				in.	out.	$\widetilde{S}_{\Pi\Pi}$	Full		
32	1 282 401	irNK	**16**	-	16	-	341	178.98s	64%
		irNL-1	19	8	11	31	252	165.35s	69%
		irNL-2	20	17	**3**	63	**71**	**114.06s**	**100%**
		irNL-3	31	**0**	4	**0**	87	163.79s	70%
		irNL-4	34	**0**	5	**0**	112	198.37s	58%
32 768	1 310 796 801	irNK	**23**	-	23	-	722	307.87s	37%
		irNL-1	26	12	14	60	472	259.02s	44%
		irNL-2	31	27	**4**	134	155	199.05s	57%
		irNL-3	30	**0**	**4**	**0**	**121**	**176.47s**	**65%**
		irNL-4	35	**0**	5	**0**	165	224.48s	51%
131 072	5 243 033 601	irNK	**24**	-	24	-	766	371.68s	31%
		irNL-1	26	12	14	60	467	316.50s	36%
		irNL-2	35	31	**4**	153	160	278.56s	41%
		irNL-3	29	**0**	**4**	**0**	**119**	**206.57s**	**55%**
		irNL-4	38	**0**	5	**0**	165	281.22s	41%

6.5 Nonlocal Nonlinearities in Two Dimensions

In the following, we also consider "Nonlocal Nonlinearities", i.e., the nonlinearities are no longer restricted to the subdomains but cross the interface. As a consequence, the elimination set $\mathcal{E}$ in NL-4 does not contain all nonlinearities. Thus, NL-4 is expected to be less efficient compared to 'Localized Nonlinearities".

In this section, we deal with channels, where each column of subdomains contains a vertical channel of width $0.5H$; see Figure 6.1 (bottom left). Since the channels do not touch subdomain vertices, the elimination set of NL-3 contains all nonlinearities. The same holds for the elimination set of NL-2 since it contains all variables.

Table 6.5: Model problem "Localized Nonlinearities" in 3D (see Section 6.1.1); nonlinear FETI-DP algorithms (NL-X, $X = 1, \dots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); domain $\Omega = [0, 1.5] \times [0, 1] \times [0, 1]$ decomposed into cubic subdomains; $p = 4$ and a weight of $\alpha = 1$ inside the inclusions and $p = 2$ and $\beta = 1$ elsewhere; $H/h = 30$; centered spherical inclusions with diameter $0.6H$; $\varepsilon_I = 10^{-5}$; $\varepsilon_O = 10^{-8}$; two MPI ranks per core; computed on magnitUDE. This table has already been published in [85]; see [85, Tab. 6].

Localized Nonlinearities in 3D $p = 4$; $H/h = 30$; **exact FETI-DP; computed on magnitUDE**								
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**Coarse Fact.**		**Krylov Its.**	**Exec. Time**	**Parallel Effic.**
				in.	**out.**			
96	2 650 021	NK	**17**	-	17	804	464.92s	65%
		NL-1	22	16	6	278	337.06s	89%
		NL-2	23	20	**3**	146	302.91s	99%
		NL-3	24	**0**	**3**	**150**	**300.41s**	**100%**
		NL-4	30	**0**	4	161	377.07s	80%
768	20 967 241	NK	**22**	-	22	805	786.32s	38%
		NL-1	27	20	7	580	480.00s	63%
		NL-2	29	25	**4**	319	441.04s	68%
		NL-3	31	**0**	**4**	308	446.72s	67%
		NL-4	36	**0**	**4**	**299**	**418.66s**	**72%**
6 144	166 811 281	NK	**27**	-	27	2 437	1 085.29s	28%
		NL-1	31	24	7	689	587.34s	51%
		NL-2	33	29	**4**	377	540.86s	56%
		NL-3	36	**0**	5	396	540.25s	56%
		NL-4	41	**0**	**4**	**344**	**490.25s**	**61%**

The results of different simulations on magnitUDE are presented in Table 6.7 and Figure 6.8, where we have used a maximum of 6 144 MPI ranks.

As expected, the performance of NL-4 is deteriorated. It turns out that it is even worse than the traditional NK approach. In contrast, the remaining variants perform similar to "Local Nonlinearities"; see Section 6.4.

As before, the cost of the coarse problem is not that relevant for the considered problem sizes. Therefore, the increased local work of NL-3 compared to NL-2 cannot be compensated by less coarse solves. Thus, NL-2 is the fastest method, but NL-3 slightly catches up. The performance of NL-2 and NL-3 are convincing since both are more than five times faster than the NK approach for the largest tests. Even the NL-1 algorithm is more than twice as fast as the NK approach for the same problem sizes.

The results presented in this section clearly indicate that the choice of the elimination set is crucial for the performance of our nonlinear FETI-DP variants, which motivates the choice of problem-dependent elimination sets; see Section 6.7.

Table 6.6: Model problem "Localized Nonlinearities" in 3D (see Section 6.1.1); nonlinear FETI-DP algorithms (NL-X, $X = 1, \dots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); we use the same settings as in Table 6.5 except for $\Omega = [0, 4] \times [0, 4] \times [0, 3]$; computed on magnitUDE. This table has already been published in [85]; see [85, Tab. 7].

Localized Nonlinearities in 3D $p = 4$; $H/h = 30$; **exact FETI-DP; computed on magnitUDE**								
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**Coarse Fact.**		**Krylov Its.**	**Exec. Time**	**Parallel Effic.**
				in.	**out.**			
48	1 332 331	NK	**13**	-	13	430	316.93s	90%
		NL-1	20	13	7	243	300.59s	95%
		NL-2	24	20	**4**	**110**	307.18s	93%
		NL-3	23	**0**	**4**	129	**286.62s**	**100%**
		NL-4	35	**0**	6	179	425.14s	67%
384	10 512 661	NK	**14**	-	14	1 002	668.84s	43%
		NL-1	22	14	8	568	543.11s	53%
		NL-2	25	21	**4**	**263**	418.15s	69%
		NL-3	23	**0**	**4**	372	**381.35s**	**75%**
		NL-4	38	**0**	6	392	565.94s	51%
3 072	83 521 321	NK	**17**	-	17	1 560	704.02s	41%
		NL-1	22	14	8	687	495.83s	58%
		NL-2	24	20	**4**	**330**	410.92s	70%
		NL-3	22	**0**	**4**	364	**377.20s**	**76%**
		NL-4	40	**0**	6	515	565.37s	51%
24 576	665 858 641	NK	**17**	-	17	1 586	1 634.88s	18%
		NL-1	23	15	8	673	988.49s	29%
		NL-2	24	20	**4**	**357**	756.77s	38%
		NL-3	23	**0**	**4**	371	**613.18s**	**47%**
		NL-4	44	**0**	6	554	967.76s	30%

6.6 Choosing the Accuracy of the Inner Newton Iteration: Numerical Results

In this section, we analyze the effectiveness of the additional control of the inner Newton iteration for our nonlinear FETI-DP methods. Therefore, we consider our model problem "Grid" of the type "Nonlocal Nonlinearities", where the nonlinearities have a more global character compared to "Channels". In addition, we consider a domain decomposition with ragged edges; see [85, Fig. 8, right] for an example of a domain decomposition with ragged edges for 16 subdomains. Since the "Grid" is an arrangement of centered vertical and horizontal channels that do not touch the domain boundary $\partial\Omega$, only the elimination sets in NL-2 and NL-3 contain all nonlinearities, but not the elimination set in NL-4. As a consequence, NL-4 is expected to be inefficient for this type of model problem.

In contrast to the previous sections, we provide results obtained from our sequential MATLAB implementation. In addition, we consider the NL-1 method without the additional computation of an initial value. In Table 6.8 we com-

Table 6.7: Model problem "Nonlocal Nonlinearities" - "Channels" (see Section 6.1.1); nonlinear FETI-DP algorithms (NL-X, $X = 1, \dots, 4$) compared to the more traditional Newton-Krylov-FETI-DP method (NK); $p = 4$ and a multiplicative weight of $\alpha = 10^5$ inside the channels and $p = 2$ and $\beta = 1$ elsewhere; each subdomain intersected by one channel; width of a channel is $0.5H$; domain $\Omega = [0, 1.5] \times [0, 1]$ decomposed into square subdomains; $H/h = 400$; $\varepsilon_I = 10^{-7}$; $\varepsilon_O = 10^{-8}$; two MPI ranks per core; computed on magnitUDE; see also Figure 6.8. This table has already been published in [85]; see [85, Tab. 8].

Nonlocal Nonlinearities - Channels **2D; $\alpha = 10^5$; $p = 4$; $H/h = 400$; exact FETI-DP; computed on magnitUDE**								
MPI Ranks	**Problem Size**	**Nonlin. Solver**	**Local Fact.**	**Coarse Fact.**		**Krylov Its.**	**Exec. Time**	**Parallel Effic.**
				in.	**out.**			
24	3 844 001	NK	**15**	-	15	420	138.48s	85%
		NL-1	21	11	10	367	139.23s	85%
		NL-2	30	26	**4**	119	**118.37s**	**100%**
		NL-3	53	**0**	**4**	**115**	183.39s	65%
		NL-4	71	**0**	12	561	328.94s	36%
96	15 368 001	NK	**13**	-	13	1 171	230.88s	51%
		NL-1	21	11	10	804	203.36s	58%
		NL-2	30	26	**4**	265	**144.75s**	**82%**
		NL-3	53	**0**	**4**	**261**	196.99s	60%
		NL-4	76	**0**	13	1 818	531.38s	22%
384	61 456 001	NK	**12**	-	12	2 553	433.40s	27%
		NL-1	19	10	9	1 193	252.97s	47%
		NL-2	29	25	**4**	426	**164.38s**	**72%**
		NL-3	44	**0**	**4**	**424**	196.16s	60%
		NL-4	62	**0**	12	3 637	775.52s	15%
1536	245 792 001	NK	**11**	-	11	4 041	692.55s	17%
		NL-1	19	10	9	1 479	304.93s	39%
		NL-2	28	24	**4**	534	**180.31s**	**66%**
		NL-3	46	**0**	**4**	**497**	217.84s	53%
		NL-4	53	**0**	12	4 596	927.91s	13%
6144	983 104 001	NK	**11**	-	11	4 698	856.28s	14%
		NL-1	19	10	9	1 666	352.16s	34%
		NL-2	24	21	**3**	427	**159.84s**	**74%**
		NL-3	31	**0**	**3**	**385**	163.69s	72%
		NL-4	42	**0**	11	4 445	937.61s	13%

pare the different NL-ane-X approaches, $X = 1, \dots, 4$, with the traditional NK-FETI-DP approach. For completeness, we also present the results of the corresponding nonlinear FETI-DP methods without the additional control of the inner Newton iteration (see the numbers in brackets). Note that "div" indicates that the corresponding method does not reach convergence.

At first, let us note that NL-1 and the traditional NK approach perform quite similar since we do not use the computation of an initial value for NL-1.

Let us further note that we do not reach convergence for NL-2 and NL-3 for and beyond 16 subdomains, which is due to an insufficient coarse space. In

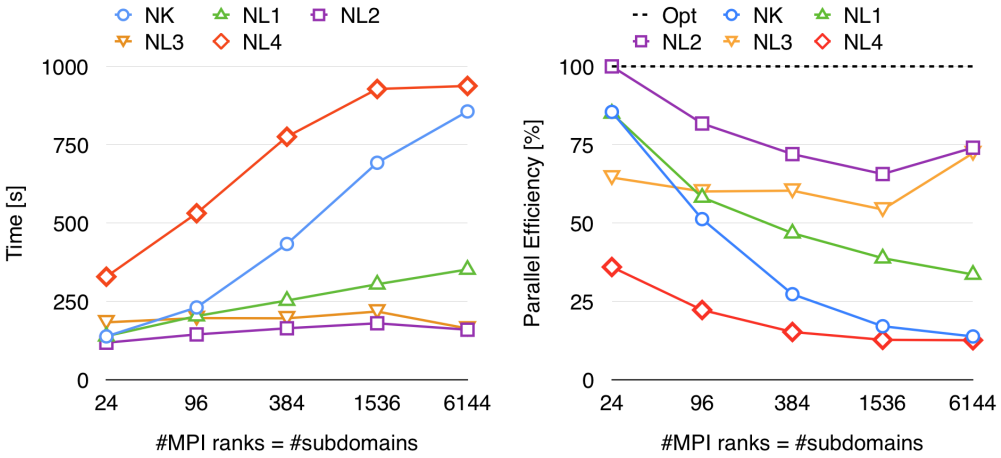


Figure 6.8: Model problem "Nonlocal Nonlinearities" - "Channels" (see Section 6.1.1); weak scalability of new FETI-DP algorithms (NL-X, $X = 1, \ldots, 4$) and the more traditional Newton-Krylov-FETI-DP algorithm (NK); performed on magnitUDE at Universität Duisburg-Essen; the fastest nonlinear FETI-DP methods (NL-2 and NL-3) are more than five times faster than the traditional NK approach; data from Table 6.7. This figure has partly been published in [85]; see [85, Fig. 12].

contrast, the corresponding NL-ane-2 and NL-ane-3 approaches yield convincing results. The NL-ane-2 approach saves more than 50 % of Newton iterations and 66 % of Krylov iterations compared to the traditional NK approach and the closely related NL-1 method without computing the initial value. Although the number of local factorizations is identical for NL-ane-2 and NL-ane-3, we obtain a higher number of outer Newton iterations for and beyond 64 subdomains for NL-ane-3. This automatically implies a higher number of Krylov iterations. Nonetheless, NL-ane-3 reduces the number of Krylov iterations by more than 50 % compared to NK and also to NL-1 without the computation of an initial value.

For NL-4, it is remarkable that the algorithm converges for all considered problem sizes in contrast to NL-2 and NL-3. However, it turns out that the outer Newton iterations and Krylov iterations are only slightly smaller compared to NK and the closely related NL-1 method without the computation of an initial value due to the fact that the nonlinearities are not completely contained in the elimination set of NL-4.

If we compare NL-4 and NL-ane-4, we obtain that the numbers of outer Newton iterations (equivalent to coarse solves) and the Krylov iterations are

Table 6.8: Model problem "Nonlocal Nonlinearities" - "Grid" (see Section 6.1.1); nonlinear FETI-DP algorithms controlling the accuracy of the inner Newton iteration (NL-ane-X, $X = 2, 3, 4$) compared to the more traditional NK-FETI-DP method (NK) and the closely related NL-1 method without computing the initial value (NL-1 no Init); numbers in brackets belong to the runs of the corresponding traditional nonlinear FETI-DP method; div indicates no convergence; $p = 4$ and a weight of $\alpha = 1$ inside the grid and $p = 2$ and $\beta = 1$ elsewhere; see also Figure 6.1 (bottom middle); $\Omega = [0, 1] \times [0, 1]$; decomposed into square subdomains; N is the number of subdomains; ragged edges (see [85, Fig. 8 (top right)]); $H/h = 16$; $\varepsilon_I = 10^{-12}$; $\varepsilon_O = 10^{-12}$; one MPI rank per core; computed on Schwarz. This table has already been published in [85]; see [85, Tab. 9].

Nonlocal Nonlinearities - Grid											
2D; $p = 4$; $H/h = 16$; exact FETI-DP; computed on Schwarz											
N	**Problem Size**	**Nonlinear Solver**	**Local Fact.**		**Coarse Fact.**				**Krylov Its.**	
					in.		**out.**			
4	1 089	NK	11		-		11		599	
		NL-1 no Init	**10**		-		10		563	
		NL-ane-2	14	(59)	7	(48)	**7**	(11)	310	(480)
		NL-ane-3	14	(76)	**0**	(0)	**7**	(14)	**307**	(805)
		NL-ane-4	14	(32)	**0**	(0)	8	(9)	425	(470)
16	4 025	NK	13		-		13		1174	
		NL-1 no Init	**12**		-		12		1148	
		NL-ane-2	15	(div)	7	(div)	**8**	(div)	490	(div)
		NL-ane-3	15	(div)	**0**	(div)	**8**	(div)	**471**	(div)
		NL-ane-4	17	(34)	**0**	(0)	10	(9)	734	(712)
64	16 641	NK	15		-		15		1891	
		NL-1 no Init	**14**		-		14		1857	
		NL-ane-2	15	(div)	9	(div)	**6**	(div)	**576**	(div)
		NL-ane-3	15	(div)	**0**	(div)	9	(div)	803	(div)
		NL-ane-4	21	(44)	**0**	(0)	13	(12)	1421	(1365)
256	66 049	NK	17		-		17		2692	
		NL-1 no Init	**16**		-		16		2602	
		NL-ane-2	18	(div)	11	(div)	**7**	(div)	**840**	(div)
		NL-ane-3	18	(div)	**0**	(div)	11	(div)	1221	(div)
		NL-ane-4	23	(51)	**0**	(0)	15	(15)	2003	(2092)

almost the same. However, we observe that the number of local factorizations decreases significantly for NL-ane-4. Note that the number of local factorizations for NL-ane-4 is less than two times the number of outer iterations, i.e., the elimination of u_I reduces the global energy J just in a few cases. As a result, the performance of NL-ane-4 is quite similar to NL-1 without the computation of an initial value. This confirms the expectation that the inner iteration is inefficient in NL-4, since not all nonlinearities are included into the elimination set.

We obtain that the additional control of the inner Newton iteration enlarges the convergence radius of nonlinear FETI-DP methods and reduces the dependency on the coarse space on one hand. On the other hand, it avoids unnecessary inner Newton iterations if the elimination set is insufficient. However, in the latter case, the NL-ane approach does not decrease outer Newton iterations and Krylov iterations in general.

6.7 A Problem-Dependent Choice of the Elimination Set in Two Dimensions

Within the ongoing joint work with Axel Klawonn, Martin Lanser, and Oliver Rheinbach on nonlinear FETI-DP methods, the design of problem-dependent nonlinear FETI-DP variants is of great interest. During the currently ongoing work on her master thesis [128], Frauke Piechulla has incorporated the problem-dependent choice of the elimination set $\mathcal{E}$ (see Section 5.4.2.5) into our existing MATLAB implementation of nonlinear FETI-DP variants. Therefore, in addition to the already existing methods NL-X, $X = 1, \dots, 4$, the software now also includes the NL-Res approach. Let us note that all presented results within this section regarding the NL-Res approach are based on the implementation of Frauke Piechulla, which, as already mentioned above, is itself based on an already existing implementation of NL-X, $X = 1, \dots, 4$. For the computation of the elimination set, we refer to Section 5.4.2.5. By construction, the size of the elimination set $\mathcal{E}$ strongly depends on the chosen parameters ρ_{res} and δ. It grows with a decreasing parameter ρ_{res} as well as with an increasing parameter δ; see Figures 6.9 and 6.10.

Within this section, we present results for different types of the model problem "Nonlocal Nonlinearities" for the p-Laplace equation; see Section 6.7.1. In addition, we also deal with a two-dimensional elasticity problem, where we take into account a Neo-Hooke material model; see Section 6.7.2. We compare different variants of NL-Res, i.e., different combinations of ρ_{res} and δ, with the remaining nonlinear FETI-DP methods as well as with the traditional NK-FETI-DP approach. Note that we consider the NL-1 method without the computation of an initial value.

In all tables within this section, we introduce the notation NL-Res(ρ_{res}) to distinguish between different variants of NL-Res. The iteration numbers resulting from different choices of δ are presented in sub-columns. Recalling that the elimination set in NL-Res may contain some primal variables, we do not know anymore whether the nonlinear elimination requires the solution of a subprob-

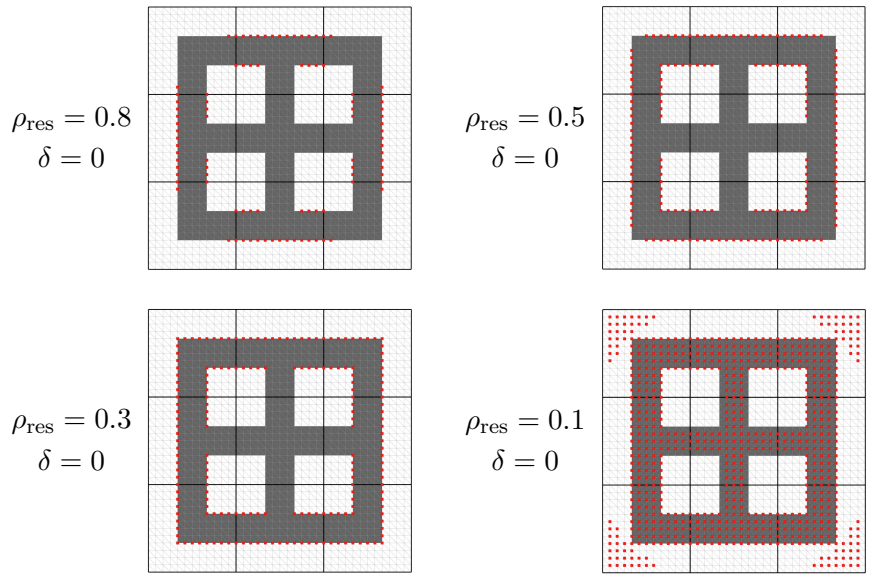


Figure 6.9: Effect of different choices of ρ_{res} on the elimination set $\mathcal{E}$ for the "Grid" considering $\delta = 0$. The finite element nodes belonging to $\mathcal{E}$ are marked with red squares; finite elements belonging to Ω_C are in dark grey.

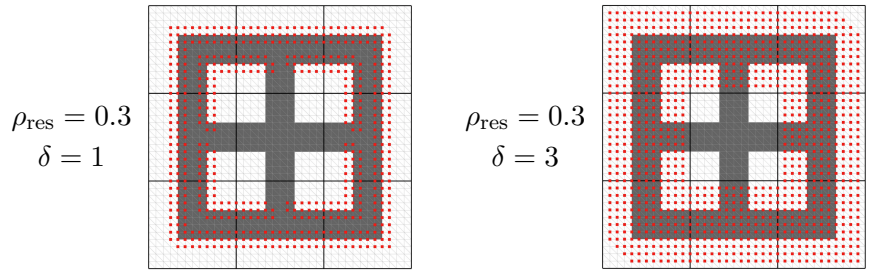


Figure 6.10: Effect of different choices of δ on the elimination set $\mathcal{E}$ for the "Grid" considering $\rho_{\text{res}} = 0.3$; see also Figure 6.9 for $\delta = 0$. The finite element nodes belonging to $\mathcal{E}$ are marked with red squares; finite elements belonging to Ω_C are in dark grey.

lem of the global coarse problem. Accordingly, we distinguish between inner and outer Newton iterations instead of local and coarse factorizations. As before, each outer Newton iteration is guaranteed to require a factorization of the coarse problem. In addition to the previous sections, we also provide the average size of the elimination sets in all tables within this section.

6.7.1 NL-Res for Nonlocal Nonlinearities

In this section, we consider the p-Laplace equation with "Nonlocal Nonlinearities". However, the nonlocal nonlinearities have a local character. On one hand, we deal with a single channel that cuts through a single column of subdomains. On the other hand, we also consider a cross of a single horizontal and a single vertical channel.

Even if the implementation enables the possibility to start with an empty elimination set in the first iteration (see Section 5.4.2.5), we do not use this option. Moreover, the elimination set is computed dynamically, i.e., it may change at the end of each outer Newton iteration. Since Newton's method converges quadratically in the neighborhood of the solution, we choose an empty elimination set whenever the norm of the nonlinear residual is at most as large as 10^{-5}. To provide a fair comparison, this strategy is also applied to NL-2, NL-3, and NL-4. Consequently, we do not have constant elimination sets for these methods anymore. As a result, the average size of the elimination set of NL-2 might be smaller compared to NL-3 and NL-4, depending on the number of overall outer Newton iterations and the number of outer Newton iterations with an empty elimination set. However, for nonempty elimination sets, the elimination set of NL-2 is larger as the elimination set of NL-3, which again is larger as the elimination set of NL-4.

For both problems considered within this section, we obtain similar results; see Tables 6.9 and 6.10. First of all, the traditional NK-FETI-DP approach and the NL-1 method without the computation of an initial value perform quite similar. It turns out that the NL-2 method performs best, but also the NL-3 and the NL-4 approach perform quite well. For the largest problems, these three methods reduce the number of outer Newton iterations and Krylov iterations by more than 50 % compared to NK. Although the number inner iterations is larger for NL-3, the number of outer iterations is identical to NL-4. It seems that the elimination of all dual variables on the interface does not have the desired effect, even if the nonlinearities cross the subdomain interface.

For the different NL-Res variants, we obtain significantly smaller average sizes of the elimination sets. Nonetheless, for the smaller problem sizes, we find at least one sufficient δ for all three parameters of ρ_{res} to achieve a performance similar to NL-3 and NL-4. However, it turns out that too small average sizes of the elimination sets do not lead to a good performance. In this case, the number of inner iterations increase, while the number of outer iterations is comparable to NK and NL-1 without the computation of an initial value. As a result, only the NL-Res(0.1) approach can compete with NL-3 and NL-4 for the largest

problem sizes of both model problems; see Tables 6.9 and 6.10. However, for a single channel, also NL-Res(0.5) performs well for $\delta > 0$ for 256 subdomains; see Table 6.9. Consequently, smaller average sizes of the elimination sets seem to be efficient for a single vertical channel, for which the nonlinearities have a more local character compared to the cross; see Tables 6.9 and 6.10. We conclude that much smaller elimination sets are sufficient to achieve a similarly good performance. Note that the increased number of inner iterations might be neglected due to the very small elimination sets.

6.7.2 NL-Res for a Contact Problem in Nonlinear Elasticity

In addition to the model problems regarding the p-Laplace equation, we also deal with nonlinear elasticity under consideration of a Neo-Hooke material model. Therefore, we consider the deformation of the unit square by a rigid tool, where we use a penalty formulation for the contact constraints; see Section 4.2.4. As before, we choose a penalty parameter $\bar{\varepsilon}_N = 500$. The initial simulation setup is presented in Figure 6.11 (left).

We do not only take into account a homogeneous material with a Young's modulus $E = 210$ and a Poisson ration $\nu = 0.3$, but also a heterogeneous material with stiffer inclusions characterized by a Young's modulus $E = 210\,000$. In both cases, the rigid tool is driven into the unit square from below. Throughout this section, the rigid tool is an arc of radius 1; see Figure 6.11 (right) for the final solution that is obtained from moving the arc 10 % in upward direction. We exclusively consider a single load step, i.e., the desired movement of the arc is applied within one step.

As in the previous section, we have compared the traditional NK-FETI-DP approach with the nonlinear FETI-DP methods NL-X, $X = 1, \ldots, 4$, as well as with different variants of the NL-Res approach, which are referred to as NL-Res(ρ_{res}) in Tables 6.11 and 6.12. For different choices of δ, the results are presented in sub-columns of the corresponding lines. Let us note that we take into account NL-1 without the additional computation of an initial value, which is therefore closely related to the traditional NK-FETI-DP method.

For all methods with nonempty elimination sets, i.e., NL-2, NL-3, NL-4, and NL-Res, we switch to the NL-1 approach whenever the current iterate is close to the solution. In the context of nonlinear elasticity, we choose empty elimination sets if the norm of the update of the displacements is not larger than 10^{-3}. In addition, we also use the option to start with a single step of NL-1 in the very first iteration.

Table 6.9: Model problem "Nonlocal Nonlinearities" - single vertical channel (see Section 6.1.1); comparison of different inner elimination sets in nonlinear FETI-DP (NL-2, NL-3, NL-4, NL-Res) to the more traditional NK-FETI-DP method (NK) and the closely related NL-1 method without computing the initial value (NL-1 no Init); $p = 4$ and a weight of $\alpha = 1$ inside the channel and $p = 2$ and $\beta = 1$ elsewhere; see also Figure 6.1 (top middle); $\Omega = [0,1] \times [0,1]$; decomposed into square subdomains; N is the number of subdomains; $H/h = 16$; $\varepsilon_I = 10^{-12}$; $\varepsilon_O = 10^{-12}$; elimination set is chosen to be empty if the norm of the nonlinear residual is not larger as 10^{-5}; channel has a width of $H/3$; one MPI rank per core; computed on Schwarz.

Nonlocal Nonlinearities - Single Vertical Channel **2D; $p = 4$; $H/h = 16$; exact FETI-DP; computed on Schwarz**														
N	**Prob. Size**	**Nonlin. Solver**	**Inner Newt. It.**			**Outer Newt. It.**			**Krylov Its.**			**∅ Size $\mathcal{E}$ [%]**		
		δ	0	1	5	0	1	5	0	1	5	0	1	5
16	4 225	NK	-			14			202			-		
		NL-1 no Init	-			13			199			0		
		NL-2	18			**5**			**72**			60		
		NL-3	32			7			100			71		
		NL-4	**23**			6			91			76		
		NL-Res(0.8)	31	38	31	9	9	7	139	141	107	0.2	0.8	4
		NL-Res(0.5)	28	40	28	7	8	6	108	123	91	0.9	2	6
		NL-Res(0.1)	24	32	30	6	7	7	89	100	100	11	12	18
64	6 641	NK	-			14			249			-		
		NL-1 no Init	-			14			264			0		
		NL-2	**22**			**5**			**87**			60		
		NL-3	28			6			105			66		
		NL-4	25			7			131			64		
		NL-Res(0.8)	49	43	33	12	9	7	223	164	127	0.07	0.3	2
		NL-Res(0.5)	35	31	28	9	7	6	168	126	108	0.3	0.9	3
		NL-Res(0.1)	27	34	32	6	7	7	108	121	122	10	10	14
256	66 049	NK	-			18			357			-		
		NL-1 no Init	-			17			363			0		
		NL-2	**24**			**5**			**99**			60		
		NL-3	35			7			129			71		
		NL-4	26			7			150			51		
		NL-Res(0.8)	64	46	40	15	9	8	309	184	160	0.03	0.1	0.5
		NL-Res(0.5)	49	28	34	11	7	7	226	143	140	0.09	0.4	0.1
		NL-Res(0.1)	36	36	37	7	7	8	134	129	148	9	9	10

For the homogeneous material, we consider a comparably large load step, where we move the arc 10 % in upward direction; see Table 6.11 for the results. It is striking that apart from some NL-Res variants only NL-4 converges for the largest problem size. Moreover, the traditional NK-FETI-DP approach already fails for a decomposition into 16 subdomains. Let us note that the divergence of NL-2 and NL-3 is related to an inappropriate coarse space.

Table 6.10: Model problem "Nonlocal Nonlinearities" - "Cross" (see Section 6.1.1); comparison of different inner elimination sets in nonlinear FETI-DP (NL-2, NL-3, NL-4, NL-Res) to the more traditional NK-FETI-DP method (NK) and the closely related NL-1 method without computing the initial value (NL-1 no Init); $p = 4$ and a weight of $\alpha = 1$ inside the cross and $p = 2$ and $\beta = 1$ elsewhere; $\Omega = [0,1] \times [0,1]$; decomposed into square subdomains; N is the number of subdomains; $H/h = 16$; $\varepsilon_I = 10^{-12}$; $\varepsilon_O = 10^{-12}$, elimination set is chosen to be empty if the norm of the nonlinear residual is not larger as 10^{-5}; horizontal and vertical channel have a width of $H/3$; one MPI rank per core; computed on Schwarz.

Nonlocal Nonlinearities - Cross **2D;** $p = 4$**;** $H/h = 16$**; exact FETI-DP; computed on Schwarz**														
N	**Prob. Size**	**Nonlin. Solver**	**Inner Newt. It.**			**Outer Newt. It.**			**Krylov Its.**			**∅ Size $\mathcal{E}$ [%]**		
		δ	0	1	5	0	1	5	0	1	5	0	1	5
16	4 225	NK	-			12			164			-		
		NL-1 no Init	-			12			175			0		
		NL-2	25			**6**			81			67		
		NL-3	35			7			94			71		
		NL-4	25			**6**			86			76		
		NL-Res(0.8)	32	42	37	11	11	7	160	159	103	0.2	0.6	5
		NL-Res(0.5)	37	39	27	9	8	**6**	129	112	132	0.6	2	9
		NL-Res(0.1)	**24**	36	25	**6**	8	**6**	82	105	**80**	9	9	20
64	6 641	NK	-			15			273			-		
		NL-1 no Init	-			15			288			0		
		NL-2	**21**			**5**			**89**			60		
		NL-3	34			7			121			71		
		NL-4	27			7			132			64		
		NL-Res(0.8)	55	54	51	15	12	9	286	232	175	0.02	0.2	1
		NL-Res(0.5)	43	55	38	11	10	7	211	193	132	0.2	0.5	3
		NL-Res(0.1)	30	38	34	7	8	7	126	140	121	5	6	10
256	66 049	NK	-			18			371			-		
		NL-1 no Init	-			17			380			0		
		NL-2	26			**5**			**99**			60		
		NL-3	37			7			128			71		
		NL-4	27			7			152			51		
		NL-Res(0.8)	63	66	69	16	14	10	352	310	219	0.01	0.04	0.3
		NL-Res(0.5)	54	66	61	13	11	10	283	246	206	0.04	0.1	0.6
		NL-Res(0.1)	**25**	36	40	**5**	7	8	100	127	154	5	4	6

As it is the case in the previous section, the NL-Res approach achieves a better performance if the average size of the elimination set is larger, i.e., the performance improves with smaller choices of ρ_{res} and larger choices of δ. It turns out that the problem-dependent choice of the elimination set is advantageous, since the average size of the elimination set is significantly smaller compared to NL-4, but can be more effective resulting in less outer iterations; see Table 6.11.

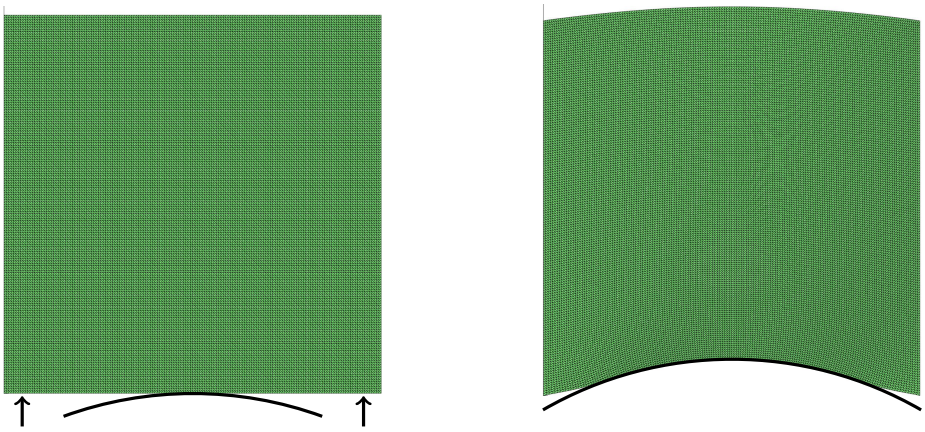


Figure 6.11: Left: Initial simulation setup for the contact problem in nonlinear elasticity. **Right:** Final solution for a homogeneous material with $E = 210$, $\nu = 0.3$ that is obtained from the NL-Res(0.1) approach with $\delta = 10$ considering a load step of 0.1 and an arc with a radius of 1.

As a result, nonlinear FETI-DP methods enable the possibility to choose larger load steps compared to the traditional NK approach. Furthermore, the NL-Res approach reduces the dependency on the coarse space compared to NL-2 and NL-3 and, therefore, enables even larger load steps. Moreover, the problem-dependent choice of the elimination set can improve the efficiency of the application of the nonlinear preconditioner.

For the heterogeneous material, we deal with random inclusions similar to the p-Laplace equation; see Figure 6.2. In this case, we have to consider a smaller load step. Otherwise, all our variants would diverge. Therefore, we half the load step size and move the arc 5 % in upward direction. For this type of model problem, we exclusively provide results for a decomposition into 64 subdomains; see Table 6.12.

First of all, it turns out that all methods reach convergence for the considered problem with a load step size of 0.05. However, the performance of NL-2 is deteriorated. It is even worse than the traditional NK approach, which is, as before, related to an inappropriate coarse space.

Especially the NL-1 method without the computation of an initial value, but also the traditional NK approach perform quite well, i.e., they need only a few number of outer iterations to reach convergence. While the nonlinear elimination in NL-3 and NL-Res does not improve the performance compared to NL-1 without the computation of an initial value, NL-4 slightly reduces the number of outer iterations and Krylov iterations. Moreover, the performance of

Table 6.11: Contact problem in nonlinear elasticity; comparison of different inner elimination sets in nonlinear FETI-DP (NL-2, NL-3, NL-4, NL-Res) to the more traditional NK-FETI-DP method (NK) and the closely related NL-1 method without computing the initial value (NL-1 no Init); simulation of a single load step with a Neo-Hooke material model; load step size: 0.1; homogeneous material with $E = 210$, $\nu = 0.3$; $\overline{\varepsilon}_N = 500$; $\Omega = [0,1] \times [0,1]$; decomposed into square subdomains; N is the number of subdomains; $H/h = 20$; $\varepsilon_I = 10^{-6}$; $\varepsilon_O = 10^{-6}$; $\mathcal{E}$ is chosen to be empty in the very first iteration as well as if the norm of the update is not larger as 10^{-3}; one MPI rank per core; computed on Schwarz.

Nonlinear Elasticity with Contact for a Homogeneous Material **2D; Neo-Hooke; $H/h = 20$; exact FETI-DP; computed on Schwarz**														
N	**Prob. Size**	**Nonlin. Solver**	**Inner Newt. It.**			**Outer Newt. It.**			**Krylov Its.**			**∅ Size $\mathcal{E}$ [%]**		
		δ	0	5	10	0	5	10	0	5	10	0	5	10
4	1 681	NK	-			8			96			-		
		NL-1 no Init	-			6			72			0		
		NL-2	8			5			60			40		
		NL-3	8			5			60			40		
		NL-4	8			5			60			38		
		NL-Res(0.5)	9	**7**	8	6	5	5	72	60	60	0.1	3	7
		NL-Res(0.3)	10	**7**	8	6	5	5	72	60	60	0.3	5	10
		NL-Res(0.1)	9	8	8	5	**4**	**4**	60	**48**	**48**	0.7	12	21
16	6 561	NK	No Conv.											
		NL-1 no Init	-			7			168			0		
		NL-2	20			7			167			43		
		NL-3	15			7			167			43		
		NL-4	11			6			143			46		
		NL-Res(0.5)	No	14	**9**	Conv.	7	6		167	142		2	3
		NL-Res(0.3)	17	10	12	7	**5**	**5**	169	**119**	118	0.3	2	6
		NL-Res(0.1)	13	11	10	6	**5**	6	145	**119**	142	0.3	5	5
64	25 921	NK	No Conv.											
		NL-1 no Init	No Conv.											
		NL-2	No Conv.											
		NL-3	No Conv.											
		NL-4	**10**			6			194			46		
		NL-Res(0.5)	No Conv.											
		NL-Res(0.3)	No	18	12	Conv.	7	**5**		228	161		0.9	2
		NL-Res(0.1)	23	**10**	11	8	6	**5**	258	193	**159**	0.1	1	3

NL-3 and all variants of NL-Res is identical, which indicates that a very small average size of the elimination set is sufficient. It is striking that the number of inner iterations in NL-Res increases with δ, i.e., with a larger elimination set. However, it has no effect on the number of outer iterations.

In addition, we also consider the NL-Res(0.1) approach without computing the elimination set anew at the end of each outer Newton iteration. It is referred to as "NL-Res(0.1) static $\mathcal{E}$" in Table 6.12. Since we start with an empty elimination set, we stick to the elimination set that is computed at the end of

Table 6.12: Contact problem in nonlinear elasticity; comparison of different inner elimination sets in nonlinear FETI-DP (NL-2, NL-3, NL-4, NL-Res) to the more traditional NK-FETI-DP method (NK) and the closely related NL-1 method without computing the initial value (NL-1 no Init); simulation of a single load step with a Neo-Hooke material model; load step size: 0.05; heterogeneous material with random inclusions; matrix material: $E = 210$, $\nu = 0.3$, inclusions: $E = 210\,000$, $\nu = 0.3$; $\bar{\varepsilon}_N = 500$; $\Omega = [0,1] \times [0,1]$; decomposed into square subdomains; N is the number of subdomains; $H/h = 20$; $\varepsilon_I = 10^{-6}$; $\varepsilon_O = 10^{-6}$; Elimination set in the very first iteration is empty for all considered methods; elimination set is chosen to be empty if the norm of the update of the displacements is not larger as 10^{-3}; one MPI rank per core; computed on Schwarz.

Nonlinear Elasticity with Contact for a Heterogeneous Material 2D; Neo-Hooke; $H/h = 20$; exact FETI-DP; computed on Schwarz															
N	**Prob. Size**	**Nonlin. Solver**	**Inner Newt. It.**			**Outer Newt. It.**			**Krylov Its.**			**∅ Size $\mathcal{E}$ [%]**		
		δ	0	5	10	0	5	10	0	5	10	0	5	10
64	25 921	NK	-			8			285			-		
		NL-1 no Init	-			6			214			0		
		NL-2	49			9			318			67		
		NL-3	15			6			213			50		
		NL-4	10			**5**			**177**			55		
		NL-Res(0.5)	**6**	11	19	6	6	6	213	214	212	0.1	2	4
		NL-Res(0.3)	9	13	19	6	6	6	213	212	212	0.2	2	4
		NL-Res(0.1)	9	12	27	6	6	6	212	211	212	0.7	6	13
		NL-Res(0.1) static $\mathcal{E}$	7	12	36	6	6	8	213	213	283	1	13	29

the first Newton iteration, as long as the norm of the update is larger than 10^{-3}.

Obviously, the dynamic choice of elimination sets is advantageous. While the average size of the elimination set is smaller, the results are at least as good as for static elimination sets; see Table 6.12. For a deeper insight, we show the evolution of the dynamically chosen elimination set $\mathcal{E}$ during the simulation of NL-Res(0.1) with $\delta = 10$. Note that the first nonempty elimination sets are identical for the static and dynamic approach, but the elimination set remains constant for the static case; see Figure 6.12 (top left).

The results indicate that the choice of elimination set affects the efficiency of the nonlinear elimination process. Consequently, the choice of dynamically chosen elimination sets is preferred. Furthermore, if we consider elasto-plastic problems, e.g., for solving boundary value problems on RVEs in FE2TI (see

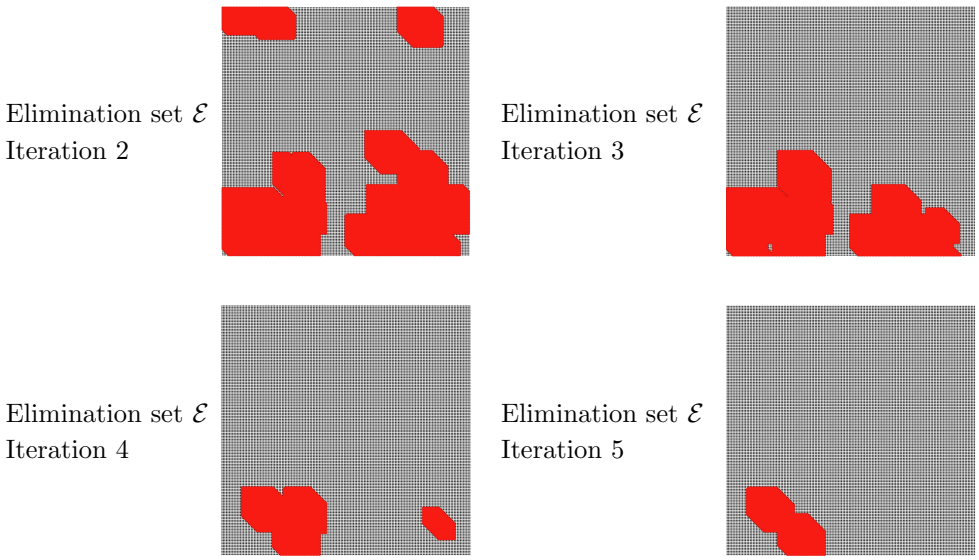


Figure 6.12: Nonempty elimination sets $\mathcal{E}$ of NL-Res(0.1) with $\delta = 10$ for random inclusions and an arc (radius 1) that moves 5 % in upward direction. Note that the elimination sets in the first and last iteration are empty.

Chapter 3), plastic yielding occurs in different regimes at different times. Therefore, the dynamic choice of the elimination set is indispensable.

6.8 Better Scalability in Nonlinear FETI-DP Methods by Localizing Work

Throughout this chapter, we have presented simulation results for different model problems and different problem sizes. If we assume that we have appropriately chosen elimination sets, all nonlinear FETI-DP methods have in common that they improve the ratio of local work and global communication as well as synchronization. As a result, they reduce computing times and improve parallel scalability.

Let us again consider "Localized Nonlinearities", since all nonlinear FETI-DP methods performed well for this type of model problem. We present the average time of an inner Newton iteration for NL-2, NL-3, and NL-4 as well as the average time of an iteration of the traditional NK-FETI-DP approach for "Standard Inclusions" computed on JUQUEEN in Figure 6.13. For further settings, we refer to the caption of Figure 6.13. Let us note that an outer Newton iteration in our nonlinear FETI-DP methods is similar to an iteration

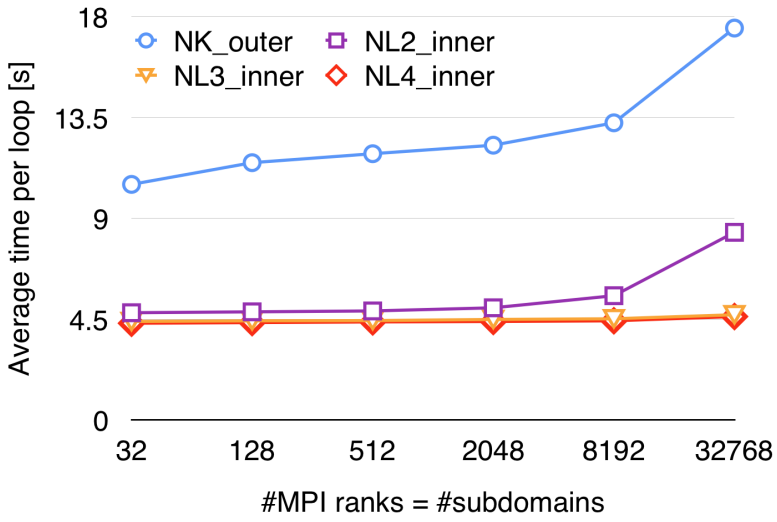


Figure 6.13: Model problem "Localized Nonlinearities" - "Standard Inclusions" (see Section 6.1.1); comparison of the average time spent in the different loops of exact nonlinear FETI-DP (NL-X, $X = 2, 3, 4$) versus Newton-Krylov-FETI-DP (NK). The inner loop (NL-2_inner, NL-3_inner, and NL-4_inner) represents the nonlinear elimination step. The timings are for $H/h = 200$; $\varepsilon_I = 10^{-6}$; $\varepsilon_O = 10^{-12}$; $\Omega = [0, 2] \times [0, 1]$; computed on JUQUEEN [69]; see also Figure 6.5. The computation of the initial value in NL-1 is not shown here, as it has a cost comparable to that of the inner loop in NL-2. This figure has already been published in [85]; see [85, Fig. 13].

of the NK approach and is therefore not shown. Furthermore, we do not present the average time of an iteration for the computation of the initial value for NL-1 since it is comparable an inner iteration for NL-2.

While we obtain convincing scalability for the inner iterations of NL-3 and NL-4, the average time per inner iteration increases for the NL-2 method for a problem size of 32 768 MPI ranks. This is related to the fact that the cost oft the exactly factorized coarse problem becomes relevant. Of course, scalability can be improved using a multilevel preconditioner instead of a sparse direct solver for the coarse problem; see Sections 5.4.4 and 6.4.2.

Obviously, the average times of the inner iterations are significantly smaller than the average time of an outer iteration due to a smaller problem size. To be more precise, we reduce the problem size at least by the Lagrange multipliers, which avoids nearest neighbor communication resulting from the application of the jump operator B. As a result, the average time of an inner Newton iteration

for NL-2, NL-3, and NL-4 is about twice as fast compared to an outer iteration. Therefore, it is valuable to increase local work in order to reduce outer Newton iterations and improve scalability.

7 Conclusion and Future Work

In this thesis, we have focused on the simulation of deformation processes of a DP steel using our highly scalable FE2TI software package. Therefore, we have presented results of several production runs. More precisely, we have simulated the Nakajima test for different sample sheet geometries in order to obtain a virtual forming limit diagram and a corresponding forming limit curve. For this purpose, we had to incorporate a contact formulation into our software package. To overcome the problem of diverging microscopic problems arising from large macroscopic deformations resulting from too large load steps, we have integrated a dynamic load step strategy. Furthermore, we have also implemented different strategies to improve the initial value of the next load step by extrapolation. In order to obtain accurate simulation results, we have taken into account an elasto-plastic material model on the microscopic level.

Fortunately, the results presented within this thesis partially exceeds the formulated goals of the EXASTEEL project. Besides the robust simulation of the Nakajima test using several different sample sheet geometries, the simulation results can be used to derive forming limit diagrams by applying the introduced evaluation strategies in a post-processing step. The corresponding forming limit curves result from linear and logarithmic interpolation for the different parts, separately, by using least squares. Although we do not have comparative experimental data, the final outcome looks quite convincing. For sure, the next step towards a virtual laboratory requires the comparison of simulation results and experimental data considering the same type of steel. Therefore, we have to ensure that we use an RVE/SSRVE that represents the material behavior of the considered type of steel in the experiment. In order to apply the FE2TI software to a wider range of contact-driven deformation processes, e.g. deep drawing, the incorporation of friction is indispensable.

Moreover, the results obtained within this thesis trigger further questions that can be considered in more detail. The simulations on the complete sample sheet indicate that the off-centered position of the final failure zone may be related to a coarse discretization. Therefore, we have to analyze the mesh dependencies on both levels, i.e., we have to perform the Nakajima test with finer discretizations and compare the results. Furthermore, we also have to

study the effect of utilizing symmetry, since we obtain differently positioned failure zones for equivalent discretizations on a symmetric quarter and on the complete sample sheet. For further comparisons of different simulations with and without exploiting symmetry, we especially have to improve the simulations on the complete sample sheet. On one hand, we may reduce the required time for a single solve of the macroscopic solution using the BDDC approach. Therefore, we have to reduce the number of Krylov iterations, which can be achieved by using an adaptive coarse space (see, e.g., [75, 91, 106, 117, 141, 142]) as well as by optimizing the decomposition into subdomains, which is obtained from METIS [72]. On the other hand, we also have to ensure that we do not require too small load steps.

In order to reduce the costs of a single simulation of the Nakajima test, which is significant for simulations on the complete sample sheet, especially for a study regarding mesh dependencies, it may be an idea not to use FE^2 on the complete sample sheet. Instead, we can use the FE^2 method in the critical area, i.e., where the failure zone is expected to evolve, and use a corresponding phenomenological material law on the macroscopic level for the remaining finite elements.

Furthermore, we can also take into account improvements regarding the RVEs to obtain a better representation of the realistic material properties. Therefore, we are going to incorporate the IVS approach (see [17, 87]), which takes into account that ferrite close to martensitic inclusions has some other properties than ferrite with a larger distance to martensitic inclusions.

Besides the simulation of the Nakajima test, we have also considered nonlinear FETI-DP methods within this thesis. Based on nonlinear right-preconditioning, which corresponds to a (partial) nonlinear elimination, we have presented a framework that covers all nonlinear FETI-DP methods. With some generalizations, we can also include a nonlinear BDDC method into the framework, which was first introduced in [85].

For the first time, we have presented a nonlinear FETI-DP method with a problem-dependent choice of the elimination set. Therefore, we use a strategy similar to that presented in [50] in the context of inexact Newton methods. The elimination set contains all degrees of freedom of a single finite element node that has a high absolute value in the nonlinear residual for at least one degree of freedom. In this context, a high value means that it is at least as large as a specific percentage of the maximum nonlinear residual. Therefore, the user provides a parameter $\rho_{\text{res}} \in [0, 1]$. In addition, the elimination set may be enlarged by including degrees of freedom of neighboring finite element

nodes. Since the resulting elimination set is based on the nonlinear residual, the corresponding nonlinear FETI-DP method is called Nonlinear-FETI-DP-Res.

In contrast to the previously introduced variants of nonlinear FETI-DP (see [85]), the Nonlinear-FETI-DP-Res approach does not stick to elimination sets that are related to the typical variable splitting in FETI-DP. By construction, the NL-FETI-DP-Res approach is expected to overcome the issues of NL-3 and NL-4, which might suffer from elimination sets that do not contain all nonlinearities, and also of NL-2, which requires the solution of the FETI-DP coarse problem in each inner iteration.

For the problems considered within this thesis, we obtain promising results. The elimination sets based on the nonlinear residual are usually much smaller compared to the elimination sets in NL-2, NL-3, and NL-4, but the resulting nonlinear elimination seems to be comparably effective. However, if the elimination sets are too small, the elimination process will become ineffective, because we spend some effort in local work without reducing global work.

Of course, we have to further analyze the Nonlinear-FETI-DP-Res approach to get a better understanding how to choose ρ_{res} and the number of neighboring finite element nodes optimally. Furthermore, we have recently implemented the NL-FETI-DP-Res approach sequentially in MATLAB. It will be exciting to see how this approach performs for larger problems in parallel. Accordingly, the NL-FETI-DP-Res method has to be incorporated into our existing parallel software. It might be also interesting to incorporate the strategy of controlling the inner Newton iteration (NL-ane) into the NL-FETI-DP-Res method. Therefore, also some further investigations regarding the NL-ane approach are required.

Furthermore, it might be interesting to consider other strategies to determine the elimination set, or to modify the implemented strategy by using the Euclidean norm instead of the infinity norm.

So far, we have only shown some local convergence properties of our nonlinear FETI-DP variants (see [85]), but it would be of great interest to have a proof, how to choose the elimination set to guarantee the improvement of the convergence of Newton's method. Moreover, the combination of our nonlinear FETI-DP variants with a globalization strategy could also be an exciting research topic.

Bibliography

[1] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM Journal on Matrix Analysis and Applications, 23(1):15–41, 2002.

[2] A. H. Baker, A. Klawonn, T. Kolev, M. Lanser, O. Rheinbach, and U. M. Yang. Scalability of classical algebraic multigrid for elasticity to half a million parallel tasks. In Software for exascale computing—SPPEXA 2013–2015, volume 113 of Lect. Notes Comput. Sci. Eng., pages 113–140. Springer, [Cham], 2016.

[3] A. H. Baker, T. V. Kolev, and U. M. Yang. Improving algebraic multigrid interpolation operators for linear elasticity problems. Numerical Linear Algebra with Applications, 17(2-3):495–517, 2010.

[4] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Web page. https://www.mcs.anl.gov/petsc, 2019.

[5] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.12, Argonne National Laboratory, 2019.

[6] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, Modern Software Tools in Scientific Computing, pages 163–202. Birkhäuser Press, 1997.

[7] D. Balzani, A. Gandhi, A. Klawonn, M. Lanser, O. Rheinbach, and J. Schröder. One-Way and Fully-Coupled FE^2 Methods for Heterogeneous Elasticity and Plasticity Problems: Parallel Scalability and an Application to Thermo-Elastoplasticity of Dual-Phase Steels. In H.-J. Bungartz, P. Neumann, and W. E. Nagel, editors, Software for Exascale Computing - SPPEXA 2013-2015, pages 91–112, Cham, 2016. Springer International Publishing.

[8] D. Balzani, L. Scheunemann, D. Brands, and J. Schröder. Construction of two- and three-dimensional statistically similar RVEs for coupled micro-macro simulations. Computational Mechanics, 54:1269–1284, 2014.

[9] J. Becker, X. Cheng, and E. Hornbogen. Dualphasen-Stähle mit erhöhter Festigkeit und Verformbarkeit. Materialwissenschaft und Werkstofftechnik, 12(9):301–308, 1981.

[10] H. Berns and W. Theisen. Eisenwerkstoffe - Stahl und Gusseisen. Springer Berlin Heidelberg, 2008.

[11] D. P. Bertsekas. Constrained optimization and Lagrange multiplier methods. Academic press, 2014.

[12] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems - scalability results on 1000 processors and discussion of highly heterogeneous problems. International Journal for Numerical Methods in Engineering, 47(1-3):513–535, 2000.

[13] F. Bordeu, P.-A. Boucard, and P. Gosselet. Balancing Domain Decomposition with Nonlinear Relocalization: Parallel Implementation for Laminates. In B. H. V. Topping and P. Iványi, editors, Proc. 1st Int. Conf. on Parallel, Distributed and Grid Computing for Engineering, Stirlingshire, UK, 2009. Civil-Comp Press.

[14] D. Braess. Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics. Cambridge University Press, 3 edition, 2007.

[15] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. Math. Comp., 50(181):1–17, 1988.

[16] D. Brands. Geometrical Modeling and Numerical Simulation of Heterogeneous Materials. PhD thesis, Universität Duisburg-Essen, 2012. Report No. 10 (2012) of the Institut für Mechanik, Universität Duisburg-Essen.

[17] D. Brands, D. Balzani, L. Scheunemann, J. Schröder, H. Richter, and D. Raabe. Computational modeling of dual-phase steels based on representative three-dimensional microstructures obtained from EBSD data. Archive of Applied Mechanics, 86(3):575–598, 2016.

[18] P. R. Brune, M. G. Knepley, B. F. Smith, and X. Tu. Composing scalable nonlinear algebraic solvers. SIAM Rev., 57(4):535–565, 2015.

[19] M. Buck, O. Iliev, and H. Andrä. Multiscale Finite Elements for Linear Elasticity: Oscillatory Boundary Conditions. In J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, editors, Domain Decomposition Methods in Science and Engineering XXI, pages 237–245, Cham, 2014. Springer International Publishing.

[20] X.-C. Cai. Nonlinear overlapping domain decomposition methods. In Domain decomposition methods in science and engineering XVIII, volume 70 of Lect. Notes Comput. Sci. Eng., pages 217–224. Springer, Berlin, 2009.

[21] X.-C. Cai and D. E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. SIAM J. Sci. Comput., 24(1):183–200 (electronic), 2002.

[22] X.-C. Cai, D. E. Keyes, and L. Marcinkowski. Non-linear additive Schwarz preconditioners and application in computational fluid dynamics. Internat. J. Numer. Methods Fluids, 40(12):1463–1470, 2002.

[23] X.-C. Cai and X. Li. Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity. SIAM J. Sci. Comput., 33(2):746–762, 2011.

[24] M. G. Cockcroft and D. J. Latham. Ductility and the workability of metals. Journal on the Institute of Metals, 48:33–39, 1968.

[25] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the LambertW function. Advances in Computational Mathematics, 5(1):329–359, 1996.

[26] J.-M. Cros. A preconditioner for the Schur complement domain decomposition method. In O. Widlund I. Herrera, D. Keyes and R. Yates, editors, Domain Decomposition Methods in Science and Engineering, pages 373–380. National Autonomous University of Mexico (UNAM), Mexico City, Mexico, ISBN 970-32-0859-2, 2003. Proc. 14th Int. Conf. Domain Decomposition Methods; http://www.ddm.org/DD14.

[27] J. Daly. A model for predicting the optimum checkpoint interval for restart dumps. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2660:3–12, 2003.

[28] T. A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. ACM Transactions on Mathematical Software, 30(2):165–195, 2004.

[29] E. A. de Souza Neto, D. Peric, and D. R. J. Owen. Computational Methods for Plasticity: Theory and Applications. Wiley, 2011.

[30] F. Devries, H. Dumontet, G. Duvaut, and F. Lene. Homogenization and damage for composite structures. International Journal for Numerical Methods in Engineering, 27(2):285–298, 1989.

[31] E. Doege and B.-A. Behrens. Handbuch Umformtechnik 3rd edn. Springer Vieweg, Berlin, Heidelberg, 2016.

[32] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput., 25(1):246–258, 2003.

[33] V. Dolean, M. J. Gander, W. Kheriji, F. Kwok, and R. Masson. Nonlinear Preconditioning: How to use a Nonlinear Schwarz Method to Precondition Newton's Method. SIAM Journal on Scientific Computing, 38(6):A3357–A3380, 2016.

[34] M. Dryja and O. B. Widlund. Schwarz methods of neumann-neumann type for three-dimensional elliptic finite element problems. Communications on Pure and Applied Mathematics, 48(2):121–155, 1995.

[35] B. Eidel and F. Gruttmann. Elastoplastic orthotropy at finite strains: multiplicative formulation and numerical implementation. Computational Materials Science, 28(3-4):732–742, 2003.

[36] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. SIAM J. Sci. Comput., 17(1):16–32, 1996. Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994).

[37] R. D. Falgout, J. E. Jones, and U. M. Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. Lecture Notes in Computational Science and Engineering, 51:267–294, 2006.

[38] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. Internat. J. Numer. Methods Engrg., 50(7):1523–1544, 2001.

[39] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. Numerical Linear Algebra with Applications, 7(7-8):687–714, 2000.

[40] C. Farhat and J. Mandel. The two-level FETI method for static and dynamic plate problems Part I: An optimal iterative solver for biharmonic systems. Computer Methods in Applied Mechanics and Engineering, 155(1):129 – 151, 1998.

[41] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. Comput. Methods Appl. Mech. Engrg., 115(3-4):365–385, 1994.

[42] C. Farhat, K. Pierson, and M. Lesoinne. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. Computer Methods in Applied Mechanics and Engineering, 184:333–374, 04 2000.

[43] F. Feyel. Multiscale FE^2 elastoviscoplastic analysis of composite structures. Computational Materials Science, 16(1):344 – 354, 1999.

[44] F. Feyel and J.-L. Chaboche. FE^2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. Computer Methods in Applied Mechanics and Engineering, 183(3):309 – 330, 2000.

[45] J. Fish, K. Shek, M. Pandheeradi, and M. S. Shephard. Computational plasticity for composite structures based on mathematical homogenization: Theory and practice. Computer Methods in Applied Mechanics and Engineering, 148(1):53 – 73, 1997.

[46] O. Forster. Analysis 2: Differentialrechnung im $\mathbb{R}^n$, gewöhnliche Differentialgleichungen. Grundkurs Mathematik. Springer Fachmedien Wiesbaden, 2017.

[47] O. Forster. Analysis 3: Maß- und Integrationstheorie, Integralsätze im $\mathbb{R}^n$ und Anwendungen. Aufbaukurs Mathematik. Springer Fachmedien Wiesbaden, 2017.

[48] G. Francfort. Homogenization and Linear Thermoelasticity. Siam Journal on Mathematical Analysis - SIAM J MATH ANAL, 14, 07 1983.

[49] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79(11):1309–1331, 2009.

[50] S. Gong and X.-C. Cai. A nonlinear elimination preconditioned inexact Newton method for heterogeneous hyperelasticity. SIAM J. Sci. Comput., 41(5):S390–S408, 2019.

[51] A. Greenbaum. Iterative methods for solving linear systems, volume 17 of Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[52] C. Groß. A unifying theory for nonlinear additively and multiplicatively preconditioned globalization strategies: Convergence Results and Examples from the field of Nonlinear Elastostatics and Elastodynamics. PhD thesis, 2009. Deutsche Nationalbibliothek.

[53] C. Groß and R. Krause. A Generalized Recursive Trust-Region Approach - Nonlinear Multiplicatively Preconditioned Trust-Region Methods and Applications. Technical Report 2010-09, Institute of Computational Science, Universita della Svizzera italiana, 03 2010.

[54] C. Groß and R. Krause. On the Globalization of ASPIN employing Trust-Region Control Strategies - Convergence Analysis and Numerical Examples. Technical Report 2011-03, Inst. Comp. Sci., Universita della Svizzera italiana, 01 2011.

[55] W. Han and B. D. Reddy. Plasticity: mathematical theory and numerical analysis, volume 9. Springer Science & Business Media, 2012.

[56] J. Hanson. Rotations in three, four, and five dimensions, 03 2011. Available online https://arxiv.org/abs/1103.5263; accessed on 19.03.2020.

[57] A. Heinlein, U. Hetmaniuk, A. Klawonn, and O. Rheinbach. The approximate component mode synthesis special finite element method in

two dimensions: Parallel implementation and numerical results. Journal of Computational and Applied Mathematics, 289:116 – 133, 2015. Sixth International Conference on Advanced Computational Methods in Engineering (ACOMEN 2014).

[58] A. Heinlein, A. Klawonn, M. Lanser, and J. Weber. A Frugal FETI-DP and BDDC Coarse Space for Heterogeneous Problems. Technical report, Universität zu Köln, December 2019.

[59] V. E. Henson and U. M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. Appl. Numer. Math., 41(1):155–177, 2002. Developments and trends in iterative methods for large systems of equations—in memoriam Rüdiger Weiss (Lausanne, 2000).

[60] M. R. Hestenes. The conjugate-gradient method for solving linear systems. In Proceedings of Symposia in Applied Mathematics. Vol. VI. Numerical analysis, pages 83–102. McGraw-Hill Book Company, Inc., New York, for the American Mathematical Society, Providence, R. I., 1956.

[61] R. Hill. A theory of the yielding and plastic flow of anisotropic metals. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 193(1033):281–297, 1948.

[62] R. Hill. The Mathematical Theory of Plasticity. Oxford Clarendon Press, 1950.

[63] G. A. Holzapfel. Nonlinear solid mechanics: a continuum approach for engineering. Nonlinear Solid Mechanics: A Continuum Approach for Engineering. Wiley, 2000.

[64] P. Hora, B. Eberle, and W. Volk. Numerical methods for a robust user-independent evaluation of Nakajima tests for the FLC determination. In B. S. Levy, D. K. Matlock, and C. J. van Tyne, editors, Material property data for more effective numerical analysis, pages 437 – 448, Golden, Colo., 2009-06-01. International Deep Drawing Research Group. International Conference IDDRG 2009; Conference Date: June 1-3, 2009; .

[65] F.-N. Hwang and X.-C. Cai. Improving robustness and parallel scalability of Newton method through nonlinear preconditioning. In Domain decomposition methods in science and engineering, volume 40 of Lect. Notes Comput. Sci. Eng., pages 201–208. Springer, Berlin, 2005.

[66] F.-N. Hwang and X.-C. Cai. A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms. Comput. Methods Appl. Mech. Engrg., 196(8):1603–1611, 2007.

[67] R. Izi, A. Konyukhov, and K. Schweizerhof. 3D frictionless contact problems with large load-steps based on the covariant description for higher

order approximation. Engineering Structures, 50:107 – 114, 2013. Engineering Structures: Modelling and Computations (special issue IASS-IACM 2012).

[68] D. Jocham. Bestimmung der lokalen Einschnürung nach linearer und nichtlinearer Umformhistorie sowie Ermittlung dehnungs- und geschwindigkeitsabhängiger Materialkennwerte. Phd thesis, Technische Universität München, 2018.

[69] Jülich Supercomputing Centre. JUQUEEN: IBM Blue Gene/Q Supercomputer System at the Jülich Supercomputing Centre. Journal of large-scale research facilities, 1(A1), 2015.

[70] Jülich Supercomputing Centre. JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre. Journal of large-scale research facilities, 5(A135), 2019.

[71] R. P. Kanwal. Generalized Functions Theory and Technique: Theory and Technique. Birkhäuser Boston, 2012.

[72] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM Journal on Scientific Computing, 20(1):359–392, 1998.

[73] A. Klawonn. Block-triangular preconditioners for saddle point problems with a penalty term. volume 19, pages 172–184. 1998. Special issue on iterative methods (Copper Mountain, CO, 1996).

[74] A. Klawonn, S. Köhler, M. Lanser, and O. Rheinbach. Computational homogenization with million-way parallelism using domain decomposition methods. Computational Mechanics, 2019.

[75] A. Klawonn, M. Kühn, and O. Rheinbach. Adaptive FETI-DP and BDDC methods with a generalized transformation of basis for heterogeneous problems. ETNA - Electronic Transactions on Numerical Analysis, 49:1–27, 01 2018.

[76] A. Klawonn, M. Lanser, P. Radtke, and O. Rheinbach. On an adaptive coarse space and on nonlinear domain decomposition. In Domain decomposition methods in science and engineering XXI, volume 98 of Lect. Notes Comput. Sci. Eng., pages 71–83. Springer, Cham, 2014.

[77] A. Klawonn, M. Lanser, and O. Rheinbach. Nonlinear FETI-DP and BDDC methods. SIAM J. Sci. Comput., 36(2):A737–A765, 2014.

[78] A. Klawonn, M. Lanser, and O. Rheinbach. EXASTEEL – Computational Scale Bridging using a FE^{2}TI approach with ex_nl/FE^2. In D. Brömmel, W. Frings, and B. J. N. Wylie, editors, JUQUEEN Extreme Scaling Workshop 2015, number FZJ-JSC-IB-2015-01, pages 15–21, 2015.

[79] A. Klawonn, M. Lanser, and O. Rheinbach. FE2TI (ex_nl/FE^2) EXASTEEL - bridging scales for multiphase steels, 2015. http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/FE2TI/_node.html.

[80] A. Klawonn, M. Lanser, and O. Rheinbach. Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations. SIAM J. Sci. Comput., 37(6):C667–C696, 2015.

[81] A. Klawonn, M. Lanser, and O. Rheinbach. FE^{2}TI: Computational Scale Bridging for Dual-Phase Steels. In IOS Series Advances in Parallel Computing, Volume 27, Parallel Computing: On the Road to Exascale; Proceedings of ParCo2015, pages 797–806, 2016. Also TUBAF Preprint: 2015-12, http://tu-freiberg.de/fakult1/forschung/preprints.

[82] A. Klawonn, M. Lanser, and O. Rheinbach. A highly scalable implementation of inexact nonlinear FETI-DP without sparse direct solvers. In Numerical mathematics and advanced applications—ENUMATH 2015, volume 112 of Lect. Notes Comput. Sci. Eng., pages 255–264. Springer, [Cham], 2016.

[83] A. Klawonn, M. Lanser, and O. Rheinbach. A nonlinear FETI-DP method with an inexact coarse problem. In Domain decomposition methods in science and engineering XXII, volume 104 of Lect. Notes Comput. Sci. Eng., pages 41–52. Springer, Cham, 2016.

[84] A. Klawonn, M. Lanser, O. Rheinbach, and M. Uran. New nonlinear FETI-DP methods based on a partial nonlinear elimination of variables. In Domain decomposition methods in science and engineering XXIII, volume 116 of Lect. Notes Comput. Sci. Eng., pages 207–215. Springer, Cham, 2017.

[85] A. Klawonn, M. Lanser, O. Rheinbach, and M. Uran. Nonlinear FETI-DP and BDDC methods: a unified framework and parallel results. SIAM J. Sci. Comput., 39(6):C417–C451, 2017.

[86] A. Klawonn, M. Lanser, O. Rheinbach, and M. Uran. On the accuracy of the inner Newton iteration in nonlinear domain decomposition. In Domain decomposition methods in science and engineering XXIV, volume 125 of Lect. Notes Comput. Sci. Eng., pages 435–443. Springer, Cham, 2018.

[87] A. Klawonn, M. Lanser, M. Uran, O. Rheinbach, S. Köhler, J. Schröder, L. Scheunemann, D. Brands, D. Balzani, A. Gandhi, G. Wellein, M. Wittmann, O. Schenk, and R. Janalík. Exasteel - towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing. In Software for Exascale Computing - SPPEXA 2016-2019. Springer LNSCE, 2020. Accepted for publication.

[88] A. Klawonn, M. H. Lanser, O. Rheinbach, G. Wellein, and M. Wittmann. Energy Efficiency of Nonlinear Domain Decomposition Methods. Technical report, Universität zu Köln, October 2018.

[89] A. Klawonn, L. F. Pavarino, and O. Rheinbach. Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains. Comput. Methods Appl. Mech. Engrg., 198(3-4):511–523, 2008.

[90] A. Klawonn, P. Radtke, and O. Rheinbach. FETI-DP methods with an adaptive coarse space. SIAM J. Numer. Anal., 53(1):297–320, 2015.

[91] A. Klawonn, P. Radtke, and O. Rheinbach. Adaptive Coarse Spaces for BDDC with a Transformation of Basis. In T. Dickopf, M. J. Gander, L. Halpern, R. Krause, and L. F. Pavarino, editors, Domain Decomposition Methods in Science and Engineering XXII, pages 301–309, Cham, 2016. Springer International Publishing.

[92] A. Klawonn and O. Rheinbach. Inexact FETI-DP methods. Internat. J. Numer. Methods Engrg., 69(2):284–307, 2007.

[93] A. Klawonn and O. Rheinbach. Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. Computer Methods in Applied Mechanics and Engineering, 196(8):1400 – 1414, 2007. Domain Decomposition Methods: recent advances and new challenges in engineering.

[94] A. Klawonn and O. Rheinbach. Highly scalable parallel domain decomposition methods with an application to biomechanics. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 90(1):5–32, 2010.

[95] A. Klawonn, O. Rheinbach, and O. B. Widlund. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. SIAM J. Numer. Anal., 46(5):2484–2504, 2008.

[96] A. Klawonn and O. Widlund. FETI and Neumann-Neumann iterative substructuring methods: Connections and new results. Communications on Pure and Applied Mathematics, 54(1):57–90, 2001.

[97] A. Klawonn and O. B. Widlund. Dual-primal FETI methods for linear elasticity. Comm. Pure Appl. Math., 59(11):1523–1572, 2006.

[98] A. Klawonn, O. B. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. SIAM J. Numer. Anal., 40(1):159–179, 2002.

[99] S. Klinkel. Theorie und Numerik eines Volumen-Schalen-Elementes bei finiten elastischen und plastischen Verzerrungen. PhD thesis, 2000.

[100] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J. Comput. Phys., 193(2):357–397, 2004.

[101] A. Konyukhov and K. Schweizerhof. Contact formulation via a velocity description allowing efficiency improvements in frictionless contact analysis. Comp. Mech., 33(3):165–173, 2004.

[102] A. Konyukhov and K. Schweizerhof. On some aspects for contact with rigid surfaces: Surface-to-rigid surface and curves-to-rigid surface algorithms. Computer Methods in Applied Mechanics and Engineering, 283:74–105, 2014.

[103] V. Kouznetsova, W. A. M. Brekelmans, and F. P. T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. Computational Mechanics, 27(1):37–48, 2001.

[104] V. Kouznetsova, M. G. D. Geers, and W. A. M. Brekelmans. Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme. International Journal for Numerical Methods in Engineering, 54(8):1235–1260, 2002.

[105] E. Kröner. Allgemeine Kontinuumstheorie der Versetzungen und Eigenspannungen. Archive for Rational Mechanics and Analysis, 4(1):273, 1959.

[106] M. J. Kühn. Adaptive FETI-DP and BDDC methods for highly heterogeneous elliptic finite element problems in three dimensions. PhD thesis, Universität zu Köln, February 2018.

[107] M. H. Lanser. Nonlinear FETI-DP and BDDC Methods. PhD thesis, Universität zu Köln, 2015.

[108] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes. An analysis of approximate nonlinear elimination. SIAM J. Sci. Comput., 17(2):538–559, 1996.

[109] E. H. Lee. Elastic-Plastic Deformation at Finite Strains. Journal of Applied Mechanics, 36:1, 1969.

[110] E. H. Lee and D. T. Liu. Finite-Strain Elastic-Plastic Theory with Application to Plane-Wave Analysis. J. of Appl. Physics, 38(1):19–27, 1967.

[111] J. Li and O. B. Widlund. FETI–DP, BDDC, and Block Cholesky Methods. Internat. J. Numer. Methods Engrg., 66(2):250–271, 2006.

[112] L. Liu and D. E. Keyes. Field-Split Preconditioned Inexact Newton Algorithms. SIAM J. on Scientific Computing, 37(3):A1388–A1409, 2015.

[113] D. G. Luenberger, Y. Ye, et al. Linear and Nonlinear Programming. Intern. Series in Operations Research and Management Science, 2016.

[114] J. Mandel. Balancing domain decomposition. Communications in Numerical Methods in Engineering, 9(3):233–241, 1993.

[115] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. Numer. Linear Algebra Appl., 10:639–659, 2003.

[116] J. Mandel, C. R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. Appl. Numer. Math., 54:167–193, 2005.

[117] J. Mandel and B. Sousedík. Adaptive Coarse Space Selection in the BDDC and the FETI-DP Iterative Substructuring Methods: Optimal Face Degrees of Freedom. In O. B. Widlund and D. E. Keyes, editors, Domain Decomposition Methods in Science and Engineering XVI, pages 421–428, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[118] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. Numer. Math., 88(3):543–558, 2001.

[119] J. E. Marsden and T. J. R. Hughes. Mathematical foundations of elasticity. Courier Corporation, 1994.

[120] MATLAB. version 8.2.0.701 (R2013b). The MathWorks Inc., Natick, Massachusetts, 2013.

[121] C. Miehe, J. Schröder, and J. Schotte. Computational homogenization analysis in finite plasticity Simulation of texture development in polycrystalline materials. Computer Methods in Applied Mechanics and Engineering, 171(3):387 – 418, 1999.

[122] J. Nocedal and S. Wright. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[123] Norm DIN EN ISO 12004-2:2008. Metallic materials – Sheet and strip – Determination of forming-limit curves – Part 2: Determination of forming-limit curves in the laboratory, 2008.

[124] J. T. Oden, K. Vemaganti, and N. Moës. Hierarchical modeling of heterogeneous solids. Computer Methods in Applied Mechanics and Engineering, 172(1):3 – 25, 1999.

[125] J. M. Ortega and W. C. Rheinboldt. Iterative solution of nonlinear equations in several variables, volume 30 of Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.

[126] J. Pebrel, C. Rey, and P. Gosselet. A nonlinear dual-domain decomposition method: Application to structural problems with damage. Inter. J. Multiscal Comp. Eng., 6(3):251–262, 2008.

[127] G. M. Phillips. Interpolation and approximation by polynomials, volume 14. Springer Science & Business Media, 2003.

[128] F. Piechulla. Residuenbasierte Eliminationsstrategien für nichtlineare FETI-DP Gebietszerlegungsverfahren. Master's thesis, Universität zu Köln, 2020. In preparation.

[129] F. Roters, P. Eisenlohr, T. R. Bieler, and D. Raabe. Crystal Plasticity Finite Element Methods in Materials Science and Engineering. John Wiley & Sons, Ltd, 2010.

[130] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput., 7(3):856–869, 1986.

[131] M. H. Sadd. Elasticity : Theory, Applications, and Numerics., volume 2nd ed. Academic Press, 2009.

[132] O. Schenk and K. Gärtner. Two-level dynamic scheduling in PARDISO: Improved scalability on shared memory multiprocessing systems. Parallel Computing, 28(2):187–197, 2002.

[133] L. Scheunemann, D. Balzani, D. Brands, and J. Schröder. Design of 3D statistically similar representative volume elements based on Minkowski functionals. Mechanics of Materials, 90:185–201, 2015.

[134] J. Schröder. Homogenisierungsmethoden der nichtlinearen Kontinuumsmechanik unter Beachtung von Instabilitäten. Institut für Mechanik (Bauwesen), Lehrstuhl I, Universität Stuttgart, Stuttgart, 2000. Habilitationsschrift.

[135] J. Schröder. A numerical two-scale homogenization scheme: The FE^2 -method. CISM International Centre for Mechanical Sciences, Courses and Lectures, 550:1–64, 2014.

[136] F. Shahzad, J. Thies, M. Kreutzer, T. Zeiser, G. Hager, and G. Wellein. CRAFT: A library for easier application-level Checkpoint/Restart and Automatic Fault Tolerance. IEEE Transactions on Parallel and Distributed Systems, 2018.

[137] J. C. Simo and T. J. R. Hughes. Computational Inelasticity. Interdisciplinary Applied Mathematics. Springer New York, 2000.

[138] R. J. M. Smit, W. A. M. Brekelmans, and H. E. H. Meijer. Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling. Computer Methods in Applied Mechanics and Engineering, 155(1):181 – 192, 1998.

[139] B. Smith, P. Bjorstad, and W. Gropp. Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, 2004.

[140] O. K. Smith. Eigenvalues of a Symmetric 3×3 Matrix. Commun. ACM, 4(4):168–, April 1961.

[141] B. Sousedík, J. Šístek, and J. Mandel. Adaptive-Multilevel BDDC and its parallel implementation. Computing, 95(12):1087–1119, 2013.

[142] N. Spillane and D. J. Rixen. Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. International Journal for Numerical Methods in Engineering, 95(11):953–990, 2013.

[143] P. Le Tallec. Numerical Homogeneisation Technique with Domain Decomposition Based a-posteriori Error Estimates. In M. Bercovier, M. J. Gander, R. Kornhuber, and O. Widlund, editors, Domain Decomposition Methods in Science and Engineering XVIII, pages 27–37, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[144] V. Tarigopula, O. S. Hopperstad, M. Langseth, A. H. Clausen, F. Hild, O.-G. Lademo, and M. Eriksson. A Study of Large Plastic Deformations in Dual Phase Steel Using Digital Image Correlation and FE Analysis. Experimental Mechanics, 48(2):181–196, Apr 2008.

[145] C. Tasan, M. Diehl, D. Yan, M. Bechtold, F. Roters, L. Schemmann, C. Zheng, N. Peranio, D. Ponge, M. Koyama, K. Tsuzaki, and D. Raabe. An Overview of Dual-Phase Steels: Advances in Microstructure-Oriented Processing and Micromechanically Guided Design. Annual Review of Materials Research, 45:391–431, 04 2015.

[146] M. Ulbrich and S. Ulbrich. Nichtlineare Optimierung. Mathematik Kompakt. Springer Basel, 2012.

[147] W. Volk. New experimental and numerical approach in the evaluation of the FLD with the FE-method. FLC-Zurich′06 conference on numerical and experimental methods in prediction of forming limits in sheet forming and tube hydroforming processes, pages 26–30, 2006.

[148] W. Volk and P. Hora. New algorithm for a robust user-independent evaluation of beginning instability for the experimental FLC determination. Int. J. Mat. Form., 4(3):339–346, 2011.

[149] G. Weber and L. Anand. Finite deformation constitutive equations and a time integration procedure for isotropic, hyperelastic-viscoplastic solids. Computer Methods in Applied Mechanics and Engineering, 79(2):173–202, 1990.

[150] O. Widlund and A. Toselli. Domain decomposition methods - algorithms and theory, volume 34. Springer, 10 2004.

[151] P. Wriggers. Computational Contact Mechanics. John Wiley & Sons, Ltd., 2002.

[152] O. C. Zienkiewicz, R. L. Taylor, and D. Fox. The Finite Element Method for Solid and Structural Mechanics. Butterworth-Heinemann, 7th edition, 2014.

[153] D. Zwillinger. Errata to CRC standard mathematical tables and formulae, 32nd edition. Online available; https://www.mathtable.com/errata/smtf32_errata.pdf; accessed on 19.03.2020.

[154] D. Zwillinger. CRC standard mathematical tables and formulae, 32nd edition. Advances in Applied Mathematics. CRC Press, 2012.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde.

Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Axel Klawonn betreut worden.

Teilpublikationen

- **Axel Klawonn, Martin Lanser, Oliver Rheinbach, Matthias Uran**,"New Nonlinear FETI-DP Methods Based on a Partial Nonlinear Elimination of Variables", Domain Decomposition Methods in Science and Engineering XXIII (Chang-Ock Lee, Xiao-Chuan Cai, David E. Keyes, Hyea Hyun Kim, Axel Klawonn, Eun-Jae Park, Olof B. Widlund, eds.), Lecture Notes in Computational Science and Engineering, vol. 116, Springer International Publishing, Cham, 2017, pp. 207-215
- **Axel Klawonn, Martin Lanser, Oliver Rheinbach, Matthias Uran**, "Nonlinear FETI-DP and BDDC Methods: A Unified Framework and Parallel Results", SIAM Journal on Scientific Computing, Vol. 39(6), pp. C417-C451
- **Axel Klawonn, Martin Lanser, Oliver Rheinbach, Matthias Uran**, "On the Accuracy of the Inner Newton Iteration in Nonlinear Domain Decomposition", Domain Decomposition Methods in Science and Engineering XXIV (Petter E. Bjøstad, Susanne C. Brenner, Lawrence Halpern, Hyea Hyun Kim, Ralf Kornbuch, Talal Rahman, Olof B. Widlund, eds.) Lecture Notes in Computational Science and Engineering, vol. 125, Springer International Publishing, Cham, 2018, pp. 435-443
- **Axel Klawonn, Martin Lanser, Matthias Uran, Oliver Rheinbach, Stephan Köhler, Jörg Schröder, Lisa Scheunemann, Dominik Brands, Daniel Balzani, Ashutosh Gandhi, Georg Wellein, Markus Wittmann, Olaf Schenk, Radim Janalík**, "EXASTEEL - Towards a virtual laboratory for the multiscale simulation of dual-phase steel using high-performance computing", Accepted for publication in the final reports of the DFG priority programme 1648 - Software for Exascale Computing - SPPEXA 2016-2019, Springer LNCSE, 2019

Ort, Datum Unterschrift (M. Uran)