NONLINEAR FETI-DP AND BDDC METHODS

INAUGURAL-DISSERTATION

ZUR
ERLANGUNG DES DOKTORGRADES
DER MATHEMATISCH-NATURWISSENSCHAFTLICHEN FAKULTÄT
DER UNIVERSITÄT ZU KÖLN

VORGELEGT VON
MARTIN HEINRICH LANSER
AUS NETTETAL

KÖLN, 2015

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

Prof. Dr. Oliver Rheinbach
(TU Bergakademie Freiberg)

Prof. Dr. Gerhard Starke
(Universität Duisburg-Essen)

Tag der mündlichen Prüfung: 26. Juni 2015

Abstract

In the simulation of deformation processes in material science the consideration of a microscopic material structure is often necessary, as in the simulation of modern high strength steels. A straightforward finite element discretization of the complete deformed body resolving the microscopic structure leads to very large nonlinear problems and a solution is out of reach, even on modern supercomputers. In homogenization approaches, as the computational scale bridging approach FE^2, the macroscopic scale of the deformed object is decoupled from the microscopic scale of the material structure. These approaches only consider the microstructure in a localized fashion on independent and parallel representative volume elements (RVEs). This introduces massive parallelism on the macroscopic level and is thus ideal for modern computer architectures with large numbers of parallel computational cores.

Nevertheless, the discretization of an RVE can still result in large nonlinear problems and thus highly scalable parallel solvers are necessary. In this context, nonlinear FETI-DP (Finite Element Tearing and Interconnecting - Dual-Primal) and BDDC (Balancing Domain Decomposition by Constraints) domain decomposition methods are discussed in this thesis, which are parallel solution methods for nonlinear problems arising from a finite element discretization. These approaches can be viewed as a strategies to further localize the computational work and to extend the parallel scalability of classical FETI-DP and BDDC methods towards extreme-scale supercomputers. Also variants providing an inexact solution of the FETI-DP coarse problem are considered in this thesis, combining two successful paradigms, i.e., nonlinear domain decomposition and AMG (Algebraic Multigrid). An efficient implementation of the resulting inexact reduced Nonlinear-FETI-DP-1 method is presented and scalability beyond 200 000 computational cores is showed.

Finally, a highly scalable FE^2 implementation using recent inexact reduced FETI-DP methods to solve the RVE problems on the microscopic level is presented and scalability on all 458 752 cores of the JUQUEEN BlueGene/Q system at Forschungszentrum Jülich is demonstrated.

Zusammenfassung

In der Simulation von Deformationsprozessen in den Materialwissenschaften ist die Berücksichtigung mikroskopisch kleiner Materialstrukturen oft essentiell. Ein gutes Beispiel dafür sind moderne Hochleistungsstähle. Eine detailgetreue Auflösung des gesamten Deformationsproblems mit Finiten Elementen in der Größenordnung der mikroskopischen Struktur würde zu Problemgrößen führen, die selbst auf den größten Supercomputern nicht zu handhaben wären. Homogenisierungsmethoden, wie das FE^2-Verfahren, entkoppeln deshalb die makroskopische Skala von der mikroskopischen und betrachten die Materialstruktur nur lokal, innerhalb unabhängiger repräsentativer Volumenelemente (RVE). Diese Methoden liefern somit einen massiven Parallelismus auf der makroskopischen Ebene und sind wie geschaffen für moderne Computerarchitekturen mit vielen parallelen Rechenkernen.

Jedoch kann auch die Diskretisierung der RVEs zu großen nichtlinearen Problemen führen, zu deren Lösung wiederum effiziente und skalierbare Verfahren notwendig sind. In diesem Zusammenhang werden nichtlineare FETI-DP (Finite Element Tearing and Interconnecting - Dual-Primal) und BDDC (Balancing Domain Decomposition by Constraints) Gebietszerlegungsverfahren betrachtet. Diese Löser nichtlinearer Finite Elemente Probleme können als Strategie angesehen werden, die vom Computer zu verrichtende Arbeit weiter zu lokalisieren und somit die Skalierbarkeit klassischer FETI-DP und BDDC Methoden zu verbessern. Es wird ebenfalls eine Variante besprochen, in der das nichtlineare FETI-DP Verfahren mit inexaktem FETI-DP kombiniert wird, was die inexakte Lösung des FETI-DP Grobgitterproblems mithilfe eines algebraischen Mehrgitterverfahrens (AMG) ermöglicht. Außerdem wird eine effiziente Implementierung dieses Verfahrens vorgestellt und Skalierbarkeit auf mehr als 200.000 Kernen gezeigt.

Zum Abschluss der Arbeit wird eine vollständige Implementierung des FE^2-Verfahrens präsentiert, in der inexakte FETI-DP Methoden zur Lösung der RVE-Probleme eingesetzt werden. Für dieses Softwarepaket wird Skalierbarkeit auf der gesamten JUQUEEN (BlueGene/Q mit 458.752 Rechenkernen im Forschungszentrum Jülich) präsentiert.

Acknowledgements

First of all, I owe special thanks to my advisor Axel Klawonn for the great opportunity to write my thesis in his group, for all the time he spent in discussions with me, and the encouragement he gave me during the last four years. I would also like to thank Oliver Rheinbach for his great support and his excellent FETI-DP implementations. I am grateful to both for reviewing this thesis.

I would also like to thank all of my former and current colleagues, namely Andreas Fischle, Sabrina Gippert, Alexander Heinlein, Martin Kühn, and Patrick Radtke. Finishing this thesis would have been impossible without their support, their hints, and all the long discussions in front of whiteboards and blackboards. I'm especially grateful to my former office mate Sabrina Gippert for being a good friend in all the years, and my current office mate Alexander Heinlein for sharing my sufferings and pain caused by C/C++ and MPI.

I would like to express my gratitude for the opportunity to spent three months at Lawrence Livermore National Laboratory LLNL), California, USA, in the computation scholar program and I am very thankful for many fruitful discussions with Ulrike Meier Yang and Tzanio Kolev during that time. I would also like to thank the team organizing the LLNL internship.

I would like to thank Satish Balay and Barry Smith, Argonne National Laboratory (ANL), Illinois, USA, for the fruitful cooperation and the PETSc team for providing this powerful piece of software.

Furthermore, I am grateful for the financial support by the German Research Foundation (DFG) through the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA). I would like to thank my colleagues and coauthors within the EXASTEEL project: Jörg Schröder, Daniel Balzani, Gerhard Wellein, Ashutosh Gandhi, and Holger Stengel. I also acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (LRZ, www.lrz.de), and also the Lawrence Livermore National Laboratory for providing computational time on Vulcan BlueGene/Q during my research stay. I'm also happy to express my thanks for the opportunity to use the Mira supercomputer at ANL. I also gratefully acknowledge the use of JUQUEEN during the Workshop on "Extreme Scaling on JUQUEEN" and the great support provided by Dirk Brömmel and Brian Wylie.

In particular, I would like to thank my parents, my sister Nadine, my brother Marc, Claudia, and my niece Carlotta for their support and some necessary distractions from math issues.

Finally, I am grateful for the encouragement through all my friends, especially

my longtime roommate Sebastian. I would like to emphasize the patience of my oldest friend Simon, who listened to many detailed mathematical descriptions without ever complaining.

Contents

List of Tables

List of Figures

1 Introduction

Finite element simulation of deformation processes in the field of structural mechanics can lead to very large and ill conditioned problems, especially if the considered material has a fine granular microstructure. This is the case in modern high strength steels combining higher strength and resistance with lower weight. During the fabrication process, including different heating and cooling phases, a characteristic structure of, e.g., martensite and austenite, embedded in a ferritic matrix material is formed, which can only be described in a scale up to 10^6 times smaller than the macroscopic scale of the deformed object. This microstructure is responsible for the great performance of modern high strength steels, but can also cause local peaks in stresses and a highly inhomogeneous material behavior. In order to detect the location of material instabilities under large forces, the microstructure cannot be neglected in realistic simulations representing experimental results. Unfortunately, a straightforward discretization with finite elements small enough to resolve the microscopic structure would lead to problem sizes demanding more than the computational power of current or even future supercomputers.

Different homogenization approaches have been introduced in the last decades separating the macroscale from the microscale; see, e.g., [14, 35, 49–52, 96, 110, 115, 127–129]. The discretized macroscopic deformation problem is small compared to a straightforward discretization resolving the microscale and the microstructure is only considered inside many localized and parallel problems based on representative volume elements (RVEs). These micro-macro simulations thus introduce a massive parallelism on the macroscale by providing independent microproblems. Nevertheless, the problems on the microscale can still be large and further levels of parallelism might be necessary. Efficient parallel solvers for nonlinear PDEs are thus a major factor driving the performance and scalability of large micro-macro simulations.

In this thesis, we consider three levels of parallelism. On the macroscopic and first level, the computational scale bridging approach FE^2 [49, 96, 110, 127–129] directly incorporating micromechanics into macroscopic simulations is used. While the macroscopic problem is discretized with comparable large finite elements, a boundary value problem resolving the microscopic structure is solved at each macroscopic Gauß integration point. The microscopic problems are based on the definition of a representative volume element (RVE). Volumetric averages of microscopic stress distributions are returned to the macroscopic level and replace a phenomenological material law. The communication between the microscopic problems in this homogenization approach is minimal, since the problems are weakly coupled through the solution of the macroscopic problem. Using a parallel solver for the independent RVEs, a second level of parallelism comes into play. Here, we suggest the usage of domain decomposition methods of the FETI-DP type (Finite Element Tearing and Interconnecting - Dual Primal) since FETI-DP methods [43, 44, 85, 87, 93, 94] are well known to be robust in problems of structural mechanics. A classical FETI-DP method has been awarded a Gordon Bell price in 2002 [10] for a structural mechanics simulation on unstructured grids. Let us remark that combinations of domain decomposition methods and homogenization approaches have already been discussed in, e.g., [19, 66, 98].

Scalability on more than 100 000 cores can finally be obtained by introducing a third level of parallelism by the incorporation of an algebraic multigrid method (AMG) on the FETI-DP coarse level. We refer to this class of three level parallel methods, combining FE^2 method with FETI-DP and AMG methods as FE^{2}TI and we will present a parallel FE^{2}TI implementation, which recently scaled up to the complete JUQUEEN BlueGene/Q machine at Forschungszentrum Jülich and qualified for the High-Q Club; see [79, 89].

The development of efficient micro-macro scale bridging methods (as FE^{2}TI) in order to simulate deformation processes of modern high strength steels is one goal of the DFG (Deutsche Forschungsgemeinschaft) project EXASTEEL [119]. The EXASTEEL project consists of five groups from applied mathematics, computer science, and material science and is one of thirteen consortia in the DFG Priority Programme 1648 and the research presented in this thesis is part of it. Although we will discuss all three levels of parallelism the FE^{2}TI method provides, the main focus of this work lies on the efficient solution of the nonlinear PDEs on the microscale. We will discuss the usage of nonlinear and linear domain decomposition approaches of FETI-DP and BDDC type. We will give a detailed overview of the theory of new nonlinear FETI-DP and BDDC methods, show scalability on more than 200 000 cores, and compare the nonlinear FETI-DP methods with some traditional approaches.

The traditional domain decomposition approach for the solution of nonlinear problems resulting from the discretization of nonlinear partial differential equations can be characterized by a geometric decomposition after linearization.

There, we solve a given discrete nonlinear problem

$$A(u) = 0 \tag{1.1}$$

by using a Newton type method $u^{(k+1)} = u^{(k)} - \alpha^{(k)} \delta u^{(k)}$ with a suitable step length $\alpha^{(k)}$. Newton methods are often a good choice because of their quadratic and thus optimal convergence rate in the neighborhood of the solution. In principle, the nonlinear methods in this thesis can also be formulated using any other nonlinear solution method. A recent overview of approaches to nonlinear solution methods and their combination can be found in [18]. In a Newton's method, in each iteration, we have to solve the linearized system $DA(u^{(k)})\, \delta u^{(k)} = A(u^{(k)})$, where $DA(u^{(k)})$ is the Jacobian or tangential matrix of A in the k-th iteration step $u^{(k)}$. In the context of domain decomposition methods, this can be done by overlapping or nonoverlapping algorithms, e.g., finite element tearing and interconnecting (FETI-1), dual-primal finite element tearing and interconnecting (FETI-DP), balancing domain decomposition by constraints (BDDC) or overlapping Schwarz. Such approaches are typically named Newton-Krylov Domain-Decomposition, e.g., Newton-Krylov-FETI-DP or Newton-Krylov-Schwarz.

Alternative approaches to the traditional domain decomposition approach can be characterized by linearization after a geometric decomposition. For a discussion "linearize first" vs. "decompose first" for fluid-structure-interaction problems see [48]. A powerful approach based on a nonlinear and overlapping domain decomposition method is the ASPIN (Additive Schwarz Preconditioned Inexact Newton) approach; see [22] and [95], where an overview on Jacobian-free Newton methods is given. In ASPIN, instead of solving (1.1) directly by a Newton method, first a nonlinear preconditioner G, constructed from a nonlinear additive Schwarz domain decomposition is applied, which finally reads $G(A(u)) = 0$. The ASPIN approach has shown to be more robust than the traditional Newton-Krylov-Schwarz approach and to be highly scalable, even for flow problems with high Reynolds numbers. An ASPIN implementation has also been successfully applied to nonlinear structural mechanics problems; see [65]. The ASPIN method can be equipped with a linear or a nonlinear coarse problem [23, 71, 72].

Nonlinear Schwarz type domain decompositions can also be used to construct a nonlinear solver instead of a preconditioner; see, e.g., [21, 40]. For a recursive trust region globalization framework applied to a nonlinear Schwarz domain decomposition, see [64].

In contrast to the above mentioned methods, the methods we are interested in are based on a nonlinear and nonoverlapping domain decomposition. Nonlinear

nonoverlapping domain decomposition methods have already been used for the special case of two subdomains in multiphysics coupling, see [32–34], and also for the coupling of a multiphase flow, see, e.g., [53, 54].
We will present and discuss versions of nonlinear FETI-DP and BDDC methods, which can also be characterized as nonlinear and nonoverlapping domain decomposition methods. These new methods are related to the nonlinear domain decomposition methods mentioned above, especially to the Nonlinear-FETI-1 domain decomposition approach introduced in [117]. As in FETI-DP [43, 44, 85, 87, 93, 94] and BDDC [26, 36, 99, 104, 105] for linear or linearized problems, the coarse spaces for our nonlinear methods are constructed from partial assembly of the finite elements. This has extended the scalability of these methods, see, e.g., [84, 86, 107, 130, 134, 135]. Let us note here that the coarse problem of the nonlinear FETI-DP method is incorporated into the nonlinear operator and not into the preconditioner. We will see later that the choice of the coarse space will therefore directly affect the Newton iteration rather than only the Krylov space iteration. As in the ASPIN approach, a nonlinear preconditioner can be used additionally, but we will not discuss such methods here.

The standard FETI-DP and BDDC methods were derived from the earlier FETI-1 [9,45–47,92] and Balancing Neumann-Neumann [41,92,103,133] iterative substructuring methods; see also [133]. A nonlinear Neumann-Neumann method was then introduced in [11].

One important building block of our nonlinear FETI-DP and BDDC methods is the solution of weakly coupled nonlinear subdomain problems. A similar algorithmic step has already been denoted nonlinear localization or nonlinear relocalization and is also a essential part of nonlinear FETI-1 methods [117] and also nonlinear Neumann-Neumann methods [11]. A simple linear/nonlinear strategy, which can be interpreted as a nonlinear localization, has been introduced in [102] for brittle material problems with strong localized nonlinearities. Nonlinear localization has also been proposed as an enhancement step for Newton-Krylov methods; see [25].

We will discuss different strategies of nonlinear dual primal FETI methods, named Nonlinear-FETI-DP-1 method (linearization first) and Nonlinear-FETI-DP-2 method (elimination first). In order to improve scalability, we will combine Nonlinear-FETI-DP-1 with the parallel strength of inexact FETI-DP domain decomposition methods introduced in [84]. Inexact reduced FETI-DP (irFETI-DP) has scaled up to 65 536 BlueGene/P cores in 2009 [87], which was a major step in improving the parallel efficiency of FETI-DP type methods since in [87] a traditional FETI-DP method could not solve linear elasticity problems beyond 16 384 cores on the same machine. We refer to the method combining nonlinear FETI-DP and irFETI-DP as irNonlinear-FETI-DP-1. This method was first introduced in [77]. In inexact reduced FETI-DP methods, parallel algebraic multigrid (AMG) methods are used on the FETI-DP coarse level and thus, as already mentioned, a further and third level of parallelism is introduced.

For highly scalable parallel algebraic multigrid solvers, see, e.g., [2], where the BoomerAMG preconditioner has shown to be parallel scalable to 100 000 cores in 2012. Scalability for another algebraic multigrid solvers to the complete JUQUEEN machine at Forschungszentrum Jülich, Germany, has recently been shown for porous media [73] using DUNE [7, 8] and for earth mantle convection [120].

Since we are concerned with problems from structural mechanics we will also discuss the usage of AMG methods in the context of linear and nonlinear elasticity problems. Classical AMG approaches designed for scalar partial differential equations always assume that the nullspace of the operator only consists of constant vectors. This is not the case in the field of structural mechanics, where also rotations do not affect the strain energy. AMG methods are iterative multigrid methods and correct the error of the current iterate by several smoothing steps and by a solution on the coarsest grid. Error vectors belonging to small or even zero eigenvalues are, in general, not corrected by the smoothing steps. Therefore, these error vectors, as, e.g., the rigid body modes, have to be handled by the coarse grid correction and thus have to be present on the coarsest level. An AMG interpolation operator reproducing the rigid body modes can be crucial for a fast convergence of AMG methods applied to elasticity problems. We will focus on the general matrix (GM) approach introduced in [3]. The GM approach has shown the ability to improve numerical convergence for linear elasticity problems with the help of an exact interpolation of the rigid body modes. Several other AMG approaches for linear elasticity problems have been suggested in the last decades, as, e.g., smoothed aggregation [16,136], unsmoothed aggregation [12,20,111,113,114], AMGe [15], element-free AMGe [67], or local optimization problems to incorporate the rigid body modes in the interpolation [37].

The remainder of the thesis is organized as follows. In Chapter 2 we first provide a brief overview of the traditional methods FETI-DP and Newton-Krylov-FETI-DP. We then introduce nonlinear FETI-DP and BDDC methods and give some remarks on their convergence behavior. A description of the algorithmic building blocks, as well as a presentation of the different approaches in form of a pseudocode are presented. Some sequential results for different model problems including the nonlinear p-Laplace operator provide an overview of the strengths and weaknesses of the different methods and show the numerical scalability of the nonlinear domain decomposition methods.

Parallel implementations and results are then provided in Chapter 3. First the inexact reduced Nonlinear-FETI-DP-1 method is introduced, combining the strength of inexact FETI-DP methods and nonlinear FETI-DP methods. A description of the parallel implementation and, once more, a presentation in form of a pseudocode is given. We finally present weak and strong scalability results on up to 262 144 cores, using different machines and considering different model problems. Again, we consider the p-Laplace equation and also provide results for nonlinear hyperelasticity problems in two and three dimensions.

In Chapter 4, we discuss the influence of the AMG method on the convergence and scalability of irFETI-DP methods. Therefore, we give a brief overview of the classical AMG approach for scalar PDEs and different AMG strategies for systems of PDEs. Then we provide a more detailed description of the global matrix (GM) approach introduced in [3], constructed to interpolate the rigid body modes exactly. Finally, a comparison of the parallel performance of the different AMG methods, used as preconditioner for irFETI-DP coarse space, is presented.

The fifth chapter provides a brief description of the FE^{2}TI method. We present a new parallel implementation using inexact reduced FETI-DP methods on the microscopic level, which automatically incorporates an AMG methods and thus a third parallel level. We will show scalability up to 1.8*M* MPI-ranks in two and three spatial dimensions.

Finally, we will summarize the content of this thesis and provide an outlook on possible and planned future work; see Chapter 6.

1.1 Linear Solvers

In all domain decomposition methods discussed in this thesis, two different types of solution methods are used, namely direct solvers and iterative Krylov subspace methods. Although these methods are well known, we will provide a brief overview of the LU decomposition, the preconditioned conjugate gradient method [61,69,123], and the generalized minimal residual method (GMRES) [61, 74, 124].

1.1.1 The LU Decomposition

One of the most common ways to solve a linear equation system

$$Au = f \tag{1.2}$$

is the computation of an LU decomposition

$$PA = LU,$$

with L is a lower triangular matrix, U is an upper triangular matrix, and P a permutes the rows of A. Let us remark that we have $A, L, U, P \in \mathbb{R}^{n \times n}$. The solution of system (1.2) is obtained in two steps by first solving

$$Ly = Pf,$$

and then solving

$$Uu = y.$$

These solves only cause a simple forward-backward substitution and have a complexity of $O(n^2)$ since both matrices L and U are triangular matrices.
The simplest way to compute an LU decomposition is the well known Gauß elimination and has, in its basic form, a complexity of $O(n^3)$. Taking the sparsity of A into account and therefore only saving nonzero entries, can significantly reduce the complexity and the memory consumption. During the elimination process the sparsity pattern can change drastically and new nonzero entries, so called *fill-in*, will appear. Too much *fill-in* can destroy the efficiency since many allocations have to be performed due to save the new entries. Here, a reordering of the rows or columns of A is necessary to reduce the *fill-in*. Effective reordering algorithms are, e.g., nested dissection [55, 101] or the minimum degree approach [56]. For a typical sparsity pattern arising from a finite element discretization of a partial differential equation, the complexity of the LU

decomposition can be reduced to $O(n^2)$.

Effective and fast sparse direct solver implementations have been presented in the last decades, as, e.g., UMFPACK [27], MUMPS [28], SuperLU [31, 100], Spooles [1], or PARDISO [97, 125, 126]. A numerical evaluation and comparison of the different packages can be found in [60]. In this thesis, we always use MUMPS or UMFPACK. We recently tested PARDISO in order to obtain a hybrid MPI/OpenMP parallel FETI-DP method; see [80].

1.1.2 The Preconditioned Conjugate Gradient Method

Let us first consider the classical conjugate gradient (CG) method without a preconditioner, introduced in [69]. The CG method is based on the minimization of the quadratic energy

$$E(u) := \frac{1}{2} u^T A u - f^T u \tag{1.3}$$

over a Krylov subspace. This minimum will be equivalent to the solution of the system

$$Au = f$$

if A is symmetric and positive definite. The CG method is an iterative method with iterates $u_{k+1} := u_k + \alpha_k p_k$ and the search directions p_K are chosen to be A-orthogonal to each other, i.e., $p_i^T A p_j = 0$ for all $i \neq j$. The iterate p_k is obtained by an orthonormalization step of the residual $r_K := Au_k - f$ against all former search directions p_i, $i < k$. In the CG method, assuming exact arithmetics, only the last two search directions have to be stored; see, e.g., [61, 123].

The preconditioned conjugate gradient method (PCG) is simply applying the CG method to the preconditioned system

$$M_1^{-1} A (M_1^{-1})^T u_1 = M_1^{-1} f,$$

with $M := M_1^T M_1$ and $u_1 = M_1^T u$. This is equivalent to the preconditioned system

$$M^{-1} A u = M^{-1} f,$$

and we obtain a convergence estimation of PCG by

$$\frac{||u_k - u^*||_A}{||u_0 - u^*||_A} \leq 2 \left(\frac{\kappa\left(M^{-1}A\right)^{\frac{1}{2}} - 1}{\kappa\left(M^{-1}A\right)^{\frac{1}{2}} + 1} \right)^k,$$

with the condition number $\kappa(\cdot) := \frac{\lambda_{max}(\cdot)}{\lambda_{min}(\cdot)}$, the exact solution u^* of $Au = f$, and the operator norm $||x||_A := (x^T A x)^{\frac{1}{2}}$. Thus, M^{-1} is a good preconditioner for

A, if $\kappa(M^{-1}A)$ is small. Additionally, matrix vector products of the kind $M^{-1}x$ should be cheap.

1.1.3 The Generalized Minimal Residual Method

As the PCG method, the generalized minimal residual method (GMRES), see [61, 124] algorithm solves linear equation systems

$$Au = f$$

by minimizing the residual over a Krylov subspace using A-orthogonal search directions p_k. Left or right preconditioners M^{-1} can also be used in GMRES methods to accelerate convergence. Opposed to the PCG approach, the matrix A can be unsymmetric and thus GMRES can be seen as a generalization of PCG. In the GMRES method the full Krylov basis has to be saved in order to perform the orthogonalization of the search directions. This results in a higher memory consumption. A truncation of the basis can be used in order to save memory, which is well known as restarted GMRES.

In general, the convergence behavior of GMRES cannot be described only by the condition number or the eigenvalues of the preconditioned matrix $M^{-1}A$. However, in several special cases an estimate is possible, as, e.g., in the case that A is symmetric positive definite, we obtain

$$\frac{||u_k - u^*||_A}{||u_0 - u^*||_A} \leq \left(\frac{2\kappa\left(M^{-1}A\right) - 1}{2\kappa\left(M^{-1}A\right)} \right)^{\frac{k}{2}},$$

which has a similar quality as the convergence estimation of the PCG method.

1.2 Newton's Method

Newton's method is an iterative method to find roots of a nonlinear function $K(u)$. We consider the minimization problem

$$\min_{u \in \mathbb{R}^n} J(u),$$

with $J : \mathbb{R}^n \to \mathbb{R}$, and the first-order necessary condition

$$K(u) := \nabla J(u) = 0.$$

Then, the Newton iteration to find a root of $K(\cdot)$ reads

$$u_{k+1} := u_k - \alpha_k \left(DK(u_k)\right)^{-1} K(u_k),$$

with the tangential matrix $DK(u_k)$ and a suitable step length α_k. If we assume that $DK(u_k)$ is positive definite, the search direction $p_k := -DK(u_k)^{-1}K(u_k)$ will be the minimizer of the second-order Taylor series of J, and thus p_k will be a decent direction.

The convergence rate of the Newton iteration u_k will be quadratic and the iterates u_k will converge to the solution u^* of $K(u) = 0$ if $DK(u^*)$ is positive definite and the initial value u_0 lies in a sufficiently small environment of u^*; see, e.g., Theorem 3.5 in [112]. To obtain global convergence, a globalization technique as a line search or a trust region method has to be used; c.f. Sections 1.3 and 1.4.

Let us remark that, in general, $DK(\cdot)$ is not positive definite. In this case the search direction might not be a decent direction with respect to the energy J. Therefore, additional effort has to be invested in order to obtain global convergence. Typically, a Quasi-Newton method, using positive definite approximations of $DK(\cdot)$ can be combined with any globalization technique; see [112].

1.3 The Line Search Approach

For an iterative method of the form

$$u_{k+1} = u_k + \alpha_k p_k$$

constructed to minimize an energy functional $J(u)$, as, e.g., Newton's method or steepest decent method (see [112]), the choice of a proper step length α_k can be crucial for a fast convergence. In general, the perfect choice would be the minimizer

$$\alpha_k := \arg \min_{\alpha \in \mathbb{R}^+} J(u_k + \alpha p_k).$$

In most cases, it is too expensive to compute this value exactly. Exceptions are iterative solvers as CG or GMRES method, where the energy E is quadratic; see equation (1.3).

Alternatively, in common line search methods, we demand α_k to fulfill weaker conditions ensuring at least a reduction of J. The simplest strategy is to fulfill the **Armijo condition**

$$J(u_k + \alpha_k p_k) \leq J(u_k) + c_1 \alpha_k \nabla J(u_k)^T p_k,$$

```
Choose initial $\alpha > 0$ and $\rho < 1$
Repeat until Armijo condition is fulfilled:
    $\alpha := \rho\alpha$
end (repeat)
$\alpha_k := \alpha$
```

Fig. 1.1: Simple backtracking algorithm to obtain a step length fulfilling the Armijo condition.

with a given constant $c_1 \in (0,1)$. Here, the reduction of J has to be proportional to the length of the step α_k and the derivative ∇J in direction p_k. The Armijo condition can be obtained with the help of a simple backtracking algorithm; see Fig. 1.1.

In the algorithms discussed during this thesis, we often consider step lengths fulfilling the Wolfe conditions in our Newton iteration. The two Wolfe conditions are stronger than the Armijo condition since they additionally take the derivative $\nabla J(u_k + \alpha_k p_k)$ into account. The **Wolfe conditions** read

$$\begin{aligned} J(u_k + \alpha_k p_k) &\leq J(u_k) + c_1 \alpha_k \nabla J(u_k)^T p_k \qquad (1.4) \\ \nabla J(u_k + \alpha_k p_k)^T p_k &\geq c_2 \nabla J(u_k)^T p_k, \qquad (1.5) \end{aligned}$$

with $0 < c_1 < c_2 < 1$. For further types of line search conditions, as, e.g., strong Wolfe conditions or Goldstein conditions, see [112].

1.4 The Trust Region Approach

Although we always use a Newton search direction combined with a line search globalization approach in our computations, we provide a brief description of the alternative trust region method. As in line search methods, we consider a quadratic approximation or a quadratic model of our nonlinear function based on Taylors series. Recall that in Newton's method the quadratic model is only used to obtain the search direction and the step length is computed afterwards. In trust region methods the direction and the step lengths are computed simultaneously regarding the quadratic model function. Since we cannot rely on the correctness of the quadratic approximation far away from the current iterate u_k, an environment of u_k is determined. In this *trust region*, we rely on our quadratic model and detect search direction and step length with the additional constraint that the new iterate has to be within the trust region. This is done by a minimization with constraints of the model problem. Depending on the quality of the new iterate, the size of the trust region is enlarged or reduced in the next step; see [112] for more details.

1.5 Numerical and Parallel Scalability

We distinguish between three different types of scalability: *numerical scalability*, *weak parallel scalability*, and *strong parallel scalability*. In general, an algorithm is *scalable* if the computational effort to obtain a solution of the same accuracy is proportional to the problem size.

In the context of the solution of nonlinear PDEs with the help of domain decomposition methods, we have, in general, two types of iterative methods and both are crucial for scalability. We always have a linearization strategy, as, e.g., Newton's method, and an iterative method in order to solve the linearized systems, as, e.g., CG or GMRES. An algorithm can only be *scalable* if the number of iterations of both methods is independent of the global problem size. For Newton's method, the number of iterations can be kept small with the help of some globalization technique, as, e.g., line search or trust region methods. In Krylov subspace methods as CG or GMRES, the number of iterations strongly depends on the condition number of the preconditioned linear system. Therefore, we can only expect *scalability* with a proper preconditioner, as, e.g., the FETI-DP preconditioner. Additionally, the condition number should only depend weakly on the diameter of the subdomains, e.g., a polylogarithmic dependency. To summarize, we call a domain decomposition method *weakly scalable* if the number of Newton- and Krylov iterations is independent of the problem size and only weakly dependent of the subdomain diameter. Let us remark that we neglect the computational effort caused by the small coarse problem in the discussion above.

A parallel implementation of an algorithm will be called *weakly parallel scalable*, if the time to solution remains constant while varying the number of computational cores and the problem size proportional to each other. In the context of domain decomposition methods, we typically keep the size of the subdomain problems and the number of subdomains on each computational core constant. In order to obtain *weak scalability*, the number of iterations has to be independent of the number of subdomains and thus *numerical scalability* is a crucial requirement.

Finally, *strong scalability* will be obtained if the time to solution for a problem of a fixed size is inversely proportional to the number of computational cores. In the context of domain decomposition methods, we typically keep the problem size of the subdomain problems and the total number of subdomains fixed. We only vary the number of subdomains on each computational core. In general, it is harder to achieve *strong parallel scalability* than *weak parallel scalability*.

2 Nonlinear FETI-DP and BDDC

The focus of this chapter lies on the discussion of several new nonlinear FETI-DP and BDDC methods to solve nonlinear partial differential equations. This chapter is partially published in [76] and [75], where these nonlinear methods have been introduced for the first time. Nonlinear FETI-DP and BDDC methods are based on a nonoverlapping domain decomposition of a nonlinear energy before linearization. We compare all suggested methods with the Newton-Krylov FETI-DP approach for which the domain decomposition is established after linearization and we present numerical results in order to analyze the numerical behavior of the different methods. These sequential computations are performed in MATLAB [109] and thus only provide a rough estimate of the computational cost. Therefore, we also present first parallel results in order to confirm our estimates. Later on, we combine our nonlinear framework with the parallel strength of inexact reduced FETI-DP, see, e.g., [87, 118], and present parallel scalability results on more than $200\,000$ computational cores; see Chapter 3.

2.1 Classical FETI-DP

Let us briefly introduce the classical FETI-DP method constructed to solve linear or linearized systems arising from a finite element discretization of a partial differential equation. The FETI-DP (Finite Element Tearing and Interconnecting - Dual Primal) method, first introduced in [43], is a domain decomposition method based on the geometric decomposition of a computational domain $\Omega \subset \mathbb{R}^d$, $d = 2,3$, into N nonoverlapping subdomains Ω_i. Thus, we have

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

where $\overline{\Omega}$ denotes the closure of Ω and each subdomain is a union of finite elements. We denote the associated local finite element spaces by W_i, the product space by $W = W_1 \times \ldots \times W_N$, and we define $\widehat{W} \subset W$ as the subspace of functions

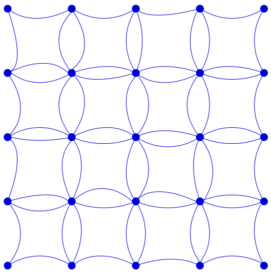

Fig. 2.1: Decomposition of a square domain Ω into 16 subdomains Ω_i. The global assembly in primal vertices denoted by Π is marked with dots. On the remaining interface variables continuity is enforced weakly with Lagrange multipliers λ, enforcing a jump condition $B_B u_B = 0$.

from W, which are continuous in all interface variables in between subdomains. We denote by

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

local restriction operators, where $V^h = V^h(\Omega)$ is the space of globally assembled finite element functions. Note that V^h and $\widehat{W}$ are isomorphic. These notations are standard in the theory of FETI-DP methods; see, e.g., [81, 93, 94, 133]. In general, all domain decomposition methods are based on the same concept, i.e., divide and conquer. Here, instead of solving a single and large problem

$$D\widehat{K}\,\hat{u} = \hat{f}$$

arising from a finite element discretization on the whole domain Ω, several smaller problems

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

associated with the subdomains Ω_i, are solved in parallel and the local solutions are reconnected on the interface. Here, the local stiffness matrices denoted by $DK^{(i)}$ and the local load vectors denoted by $f^{(i)}$ arise from a finite element assembly on the subdomains Ω_i, $i = 1, ..., N$. Since we predominantly regard nonlinear problems, we decided to denote linear and linearized systems always with DK instead of K, which is the most common notation in the linear theory. This choice ensures a consistent notation throughout all methods and chapters since $DK(\cdot)$ often denotes the tangent of a nonlinear function $K(\cdot)$. The continuity of the solution on the interface

$$\Gamma := \bigcup_{i=0}^{N} \partial\Omega_i \setminus \partial\Omega \tag{2.1}$$

is enforced by a combination of a global subassembly in some variables and additional dual conditions on the remaining variables. Therefore, we partition the degrees of freedom on Γ into sets of primal and dual variables, denoted by Π and Δ, and define the set of interior and dual variables $B := [I, \Delta]$. Here, I denotes the set of all variables which are contained in the interior part of the subdomains. Thus, the local stiffness matrix reads in block form

$$DK^{(i)} := \begin{bmatrix} DK_{BB}^{(i)} & DK_{B\Pi}^{(i)} \\ DK_{\Pi B}^{(i)} & DK_{\Pi\Pi}^{(i)} \end{bmatrix}, \text{ with } DK_{BB}^{(i)} := \begin{bmatrix} DK_{II}^{(i)} & DK_{I\Delta}^{(i)} \\ DK_{\Delta I}^{(i)} & DK_{\Delta\Delta}^{(i)} \end{bmatrix}. \tag{2.2}$$

The global assembly in primal variables Π is performed by means of the standard FETI-DP partial assembly operator R_Π^T; see, e.g., [87, 133]. The assembled system in the primal variables is also named FETI-DP coarse problem and can include more than just vertex constraints; e.g., [93]. The FETI-DP coarse problem is a global problem and in general all subdomains contribute primal constraints. Nevertheless, the coarse problem is up to 3 or 4 magnitudes smaller compared to the full problem, and computations can also be parallelized in inexact FETI-DP variants; see Chapter 3. To obtain continuity in the remaining interface variables we introduce the jump operator B_B, cf. the definition in [87, 133], and Lagrange multipliers λ to enforce the continuity condition $B_B u_B = 0$ on the variables u_B. A row in $B_B u_B = 0$ enforces equality of two variables belonging to the same physical point and quantity but two different subdomains. A typical row in

$$B_B = \left[B_B^{(1)}, \cdots, B_B^{(N)} \right]$$

thus contains only a single 1 and a single -1 and multiplications with B_B only cause nearest neighbor communication. An illustration of a typical FETI-DP decomposition can be found in Fig. 2.1.

All this leads to the FETI-DP master system

$$\begin{bmatrix} DK_{BB} & D\widetilde{K}_{\Pi B}^T & B_B^T \\ D\widetilde{K}_{\Pi B} & D\widetilde{K}_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}. \tag{2.3}$$

Here, DK_{BB} is a block diagonal matrix

$$DK_{BB} := \begin{bmatrix} DK_{BB}^{(1)} & & \\ & \ddots & \\ & & DK_{BB}^{(N)} \end{bmatrix},$$

and the diagonal blocks $DK_{BB}^{(i)}$ are the restrictions of the local matrix $DK^{(i)}$ to the set of dual and interior variables and are thus local to the subdomains. Of course, the vectors f_B and u_B have an equivalent block structure. The matrices including primal variables Π are, as already mentioned, partially assembled in the primal variables and thus global but small. It is also common in FETI-DP notation to mark the primal assembled matrices and vectors with a tilde, as, e.g., $D\widetilde{K}_{\Pi\Pi}$.

Elimination of u_B and $\tilde{u}_\Pi$ in system (2.3) leads to

$$F\lambda = d, \tag{2.4}$$

$$\text{where} \qquad F = B_B DK_{BB}^{-1} B_B^T + B_B DK_{BB}^{-1} \widetilde{K}_{\Pi B}^T \widetilde{S}_{\Pi\Pi}^{-1} \widetilde{K}_{\Pi B} DK_{BB}^{-1} B_B^T \tag{2.5}$$

$$\text{and} \qquad d = B_B DK_{BB}^{-1} f_B + B_B K_{BB}^{-1} D\widetilde{K}_{\Pi B}^T \widetilde{S}_{\Pi\Pi}^{-1} (\tilde{f}_\Pi - \widetilde{K}_{\Pi B} DK_{BB}^{-1} f_B). \tag{2.6}$$

Here,

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

is the Schur complement on the primal variables. Due to its block structure, the inversion of DK_{BB} only consists of local operations $(DK_{BB}^{(i)})^{-1}$. Finally, the FETI-DP method is the iterative solution of the preconditioned system

$$M^{-1} F\lambda = M^{-1} d \tag{2.7}$$

using a Krylov subspace method such as CG [69] or GMRES [61,124]; see also Sections 1.1.2 and 1.1.3. In this thesis, we always use the standard Dirichlet preconditioner

$$M_{FETI_D}^{-1} =: M^{-1}, \tag{2.8}$$

which is a weighted sum

$$M_{FETI_D}^{-1} := \sum_{i=1}^{N} B_{\Delta,D}^{(i)} S_{\Delta\Delta}^{(i)} B_{\Delta,D}^{(i)T} \tag{2.9}$$

of local Schur complements

$$S_{\Delta\Delta}^{(i)} := DK_{\Delta\Delta}^{(i)} - DK_{\Delta I}^{(i)} (DK_{II}^{(i)})^{-1} DK_{\Delta I}^{(i)T}$$

based on the decomposition of the local tangential or stiffness matrix depicted in (2.2). Here, $B_{\Delta,D}$ is a scaled and restricted version of the jump operator B_B to the dual variables Δ; see, e.g., [87,133] for a complete notation and different choices of weights. Let us remark that there are several other choices for M^{-1}, e.g., the lumped preconditioner

$$M^{-1}_{FETI_L} := \sum_{i=1}^{N} B^{(i)}_{\Delta,D} K^{(i)}_{\Delta\Delta} B^{(i)T}_{\Delta,D}, \tag{2.10}$$

where only the stiffness matrix on the interface is considered.
The application of M^{-1} is embarrassingly parallel and, choosing $M^{-1} := M^{-1}_{FETI_D}$, the polylogarithmic condition number bound

$$\kappa(M^{-1}F) \leq C\left(1+\log\left(\frac{H}{h}\right)\right)^2 \tag{2.11}$$

has been shown for several model problems in two and three dimensions; see [82, 90, 93, 94, 108] for the different model problems. Here, H is the typical diameter of a subdomain, h is the typical diameter of a finite element, and C is a constant independent of H, h, and jumps in PDE coefficients. Of course, in all cases sufficient and problem dependent primal constraints Π and proper weights for the preconditioner induced by the PDE coefficients have to be chosen.
Defining $\tilde{u} := [u_B^T\,,\, \tilde{u}_\Pi^T]^T$, $\tilde{f} = [f_B^T\,,\, \tilde{f}_\Pi^T]^T$, $B := [B_B\; 0]$, and

$$D\widetilde{K} := \begin{bmatrix} DK_{BB} & D\widetilde{K}^T_{\Pi B} \\ D\widetilde{K}_{\Pi B} & D\widetilde{K}_{\Pi\Pi} \end{bmatrix} \tag{2.12}$$

we can compress the blocksystem (2.3) and obtain

$$\begin{bmatrix} D\widetilde{K} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ 0 \end{bmatrix}. \tag{2.13}$$

Using this compact notation, equation 2.4 also reads

$$B(D\widetilde{K}^{-1})B^T\lambda = B(D\widetilde{K}^{-1})\tilde{f}. \tag{2.14}$$

We predominantly use the compact notations in the remainder of this thesis. Let us remark that, recalling R_Π, we can also reads

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

with the block system

$$DK = \begin{bmatrix} DK^{(1)} & & \\ & \ddots & \\ & & DK^{(N)} \end{bmatrix}. \tag{2.15}$$

2.2 Newton-Krylov-FETI-DP

A common and effective way to solve a discretized, nonlinear equation $A(\hat{u}) = 0$ is the linearization with Newton's method; cf. Section 1.2. The linearized system can then be solved with an iterative method, as, e.g., a domain decomposition method. We refer to these methods as Newton-Krylov-DD and give a detailed description of the variant Newton-Krylov-FETI-DP in this section. To obtain a consistent notation in all of our methods we first reformulate the problem $A(\hat{u}) = 0$ operating on V^h, using the nonoverlapping decomposition of Ω, to

$$R^T K(R\hat{u}) - R^T f = 0, \tag{2.16}$$

where $A(\hat{u}) = R^T K(R\hat{u}) - R^T f$, $R = \left[R_1^T, ..., R_N^T\right]^T$,

$$K(u) := \begin{bmatrix} K_1(u_1) \\ \vdots \\ K_N(u_N) \end{bmatrix}, \text{ and } f := \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}. \tag{2.17}$$

Here, $K_i(u_i)$, f_i and R_i correspond to the subdomain Ω_i, where $K_i(u_i) = f_i$ is the discretized nonlinear problem living on Ω_i; see also Section 2.4 for more details. Next, we linearize and obtain the Newton iteration

$$\hat{u}^{(k+1)} = \hat{u}^{(k)} - \alpha^{(k)} \delta\hat{u}^{(k)}, \tag{2.18}$$

with a suitable step length $\alpha^{(k)}$. In our numerical experiments, we often use a line search with the step length $\alpha^{(k)}$ satisfying the Wolfe conditions; see [112] and Section 1.3. Let us briefly recall that for a minimizing problem $\min_{u\in\mathbb{R}^n} J(u)$ and a search direction δu the Wolfe conditions read

$$J(u + \alpha\delta u) \leq J(u) + c_1 \alpha \delta u^T \nabla J(u) \tag{2.19}$$

and

$$\delta u^T \nabla J(u + \alpha\delta u) \geq c_2 \delta u^T \nabla J(u) \tag{2.20}$$

with $0 < c_1 < c_2 < 1$.

To obtain the update $\delta\hat{u}^{(k)}$, we have to solve the fully assembled linearized system

$$R^T DK(R\hat{u}^{(k)})R\delta\hat{u}^{(k)} = R^T K(R\hat{u}^{(k)}) - R^T f.$$

Here, $DK(R\hat{u}^{(k)})$ is a block diagonal matrix with the subdomain associated tangential matrices $DK^{(i)}(u_i^{(k)})$ on the diagonal and $u_i^{(k)} := R_i\hat{u}^{(k)}$ is the restriction of $\hat{u}^{(k)}$ to the subdomain Ω_i. In order to solve the linearized problem with a FETI-DP method we again introduce a set of primal variables Π and the corresponding jump operator B, acting on the remaining dual interface variables. We then solve

$$\begin{bmatrix} R_\Pi^T DK(R_\Pi\tilde{u}^{(k)})R_\Pi & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} R_\Pi^T K(R_\Pi\tilde{u}^{(k)}) - R_\Pi^T f \\ 0 \end{bmatrix}, \tag{2.21}$$

where $R_\Pi^T DK(R_\Pi\tilde{u})R_\Pi$ is the partially assembled form of the local tangential matrices and $R_\Pi^T f =: \tilde{f}$ the partially assembled right hand side. Introducing the notations

$$D\widetilde{K}(\tilde{u}) := R_\Pi^T DK(R_\Pi\tilde{u})R_\Pi \tag{2.22}$$

and

$$\widetilde{K}(\tilde{u}) := R_\Pi^T K(R_\Pi\tilde{u}), \tag{2.23}$$

we write (2.21) also as

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} \\ 0 \end{bmatrix}, \tag{2.24}$$

which is equivalent to formulation (2.13) in FETI-DP for linear equations.

The partially assembled variables $\tilde{u}$ and the assembled variables $\hat{u}$ are related to each other in the usual canonical way. Formally, we can introduce corresponding operators, i.e., we can write $\tilde{u}^{(k)} := R_{\Pi,D}^T R\hat{u}^{(k)}$ and, after solving, $\delta\hat{u}^{(k)} := R_D^T R_\Pi \delta\tilde{u}^{(k)}$. The matrix $R_{\Pi,D}^T$ is a scaled version of R_Π^T, i.e., the rows in R_Π^T which correspond to a primal variable are scaled with the inverse of the multiplicity of the primal variable. The matrix R_D^T is a scaled version of the matrix R^T, where the rows are scaled with the inverse of the multiplicity of the corresponding global degree of freedom. By one step of block elimination, we obtain from (2.24)

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ 0 & -B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} \\ -B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f}) \end{bmatrix}. \tag{2.25}$$

It remains to solve the reduced system for the Lagrange multipliers

$$F_{NK}(\tilde{u}^{(k)})\lambda = d_{NK}(\tilde{u}^{(k)}), \tag{2.26}$$

where

$$\begin{aligned} F_{NK}(\tilde{u}^{(k)}) &= B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T, \\ d_{NK} &= B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f}). \end{aligned}$$

This can again be solved by some preconditioned Krylov iteration using one of the standard FETI-DP preconditioners M^{-1}, e.g., the Dirichlet preconditioner defined in Section 2.1 or [133].

Finally, we may formulate an equivalent Newton iteration operating on $\widehat{W}$. Remember that $\widehat{W}$ is isomorphic to V^h and therefore we simply have

$$u^{(k+1)} = u^{(k)} - \alpha^{(k)}\delta u^{(k)},$$

where $u^{(k)} \in \widehat{W}$, $u^{(k+1)} \in \widehat{W}$, and $\delta u^{(k)} \in \widehat{W}$. The Newton update $\delta u^{(k)} = R_\Pi \delta \tilde{u}^{(k)}$ is obtained by solving (2.24) with $\tilde{u}^{(k)} := R_{\Pi,D}^T u^{(k)}$.

2.3 Nonoverlapping and Nonlinear Domain Decomposition

In this chapter, we describe the idea of decomposing a nonlinear energy function in general and derive a nonlinear FETI-DP master system presented in (2.34). Let us remark that all our nonlinear FETI-DP methods are based on this nonlinear master system. First, we recall the decomposition $\Omega_i, i = 1, \ldots, N$ of $\Omega \subset \mathbb{R}^d$, $d = 2,3$, into nonoverlapping subdomains.

Instead of the solution of the discrete nonlinear problem (1.1), we consider the minimization of the related nonlinear energy functional $J : V^h \to \mathbb{R}$,

$$\min_{\hat{u} \in V^h} J(\hat{u}). \tag{2.27}$$

We make the following assumption.

Assumption 1 *There exist local energy functionals $J_i : W_i \to \mathbb{R}, i = 1, \ldots, N$, such that for $\hat{u} \in V^h$, the global energy functional can be represented as a sum of local energies*

$$J(\hat{u}) = \sum_{i=1}^{N} J_i(u_i),$$

where $u_i := R_i \hat{u}$.

Assumption 1 is satisfied for the p-Laplace problem as well as for relevant nonlinear problems such as standard or incompressible hyperelasticity. In the remainder of this thesis we consider both, the p-Laplace equation and nonlinear hyperelasticity, as model problems. In general, Assumption 1 is satisfied for standard problems discretized by finite elements. Assumption 1 then simply follows from the additivity of the integral. It may not be satisfied for problems with nonlocal phenomena such as nonlocal damage models in structural mechanics.

Let us define the jump operator $B = [B_1, ..., B_N]$ which enforces continuity across the interface Γ given as the union of the interior subdomain boundaries. Here, for $u = [u_1^T, ..., u_N^T]^T$ with $u_i \in W_i$, $Bu = 0$ corresponds to $u \in \widehat{W}$. Let us remark that the jump operator B introduced in Section 2.1 enforcing continuity in the dual variables u_Δ can be obtained from B in this section by removing all lines corresponding to primal variables from the set Π. We reuse the notation here for simplicity.

Using Assumption 1 and the notation $u = [u_1^T, \ldots, u_N^T]^T \in W$, we obtain

$$\min_{\hat{u} \in V^h} J(\hat{u}) = \min_{\substack{\hat{u} \in V^h \\ u_i = R_i \hat{u}}} \sum_{i=1}^{N} J_i(u_i) = \min_{u \in \widehat{W}} \sum_{i=1}^{N} J_i(u_i) = \min_{\substack{u \in W \\ Bu=0}} \sum_{i=1}^{N} J_i(u_i). \tag{2.28}$$

We introduce the space of Lagrange multipliers $V :=$ range(B). Now, it is possible to derive a nonlinear saddle point problem.

We have to compute the stationary points of the Lagrange function

$$\begin{aligned} &\mathscr{L} : W \times V \to \mathbb{R} \\ &\mathscr{L}(u, \lambda) = \textstyle\sum_{i=1}^{N} J_i(u_i) + (Bu)^T \lambda. \end{aligned} \tag{2.29}$$

which are the solutions of the equation

$$\begin{bmatrix} \sum_{i=1}^{N} J_i'(u_i)(v_i) + (Bv)^T \lambda \\ (Bu)^T \mu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \forall v \in W, \forall \mu \in V. \tag{2.30}$$

For each $i = 1, \ldots, N$, we denote by $\varphi_{i,j}, j = 1, \ldots, N_i$, the N_i nodal finite element basis functions for the local finite element space W_i. We assume that for each $i = 1, \ldots, N$, we have the following representation $J_i'(u_i)(\varphi_{i,j}) = (K_i(u_i) - f_i)_j$. Here, f_i is independent of u_i. With using the notation

$$K(u) := \begin{bmatrix} K_1(u_1) \\ \vdots \\ K_N(u_N) \end{bmatrix}, \quad f := \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}, \quad \text{and} \quad u := \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \tag{2.31}$$

from (2.17) we obtain from (2.30) the discrete nonlinear system of equations

$$\begin{array}{lcl} K(u) + B^T\lambda & = & f \\ Bu & = & 0, \end{array} \tag{2.32}$$

which can be seen as nonlinear analogon of the linear FETI master system.

2.4 Nonlinear FETI-DP Methods

Let us recall the definition of the jump operator B from Section 2.1 which enforces continuity in all dual variables u_Δ but not in the primal variables $\tilde{u}_\Pi$.

To construct our nonlinear FETI-DP methods we use partial assembly of $K(u)$ and f from equation (2.32) in the primal variables. All our nonlinear FETI-DP methods are then based on the following nonlinear FETI-DP master system

$$\begin{array}{lcl} R_\Pi^T K(R_\Pi \tilde{u}) + B^T\lambda - R_\Pi^T f & = & 0 \\ B\tilde{u} & = & 0, \end{array} \tag{2.33}$$

where $\tilde{u} \in \widetilde{W}$, and the Lagrange multipliers $\lambda \in V$. We denote the space of finite element functions in W, which are continuous in all primal variables, by $\widetilde{W}$. Note that we have $\widehat{W} \subset \widetilde{W} \subset W$. Using the notation from (2.23) we will also write (2.33) in compact form

$$\begin{array}{lcl} \widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f} & = & 0 \\ B\tilde{u} & = & 0, \end{array} \tag{2.34}$$

where again $\tilde{f} := R_\Pi^T f$. From the chain rule and recalling definition (2.22), we have

$$D(\widetilde{K}(\tilde{u})) = D(R_\Pi^T K(R_\Pi \tilde{u})) = R_\Pi^T DK(R_\Pi \tilde{u}) R_\Pi = D\widetilde{K}(\tilde{u}), \tag{2.35}$$

i.e., the derivative of the partially assembled nonlinear operator $\widetilde{K}$ can be computed by partially assembling DK. This is equivalent to a primal assembly of the local derivatives $DK^{(i)}$ of the local subdomain operators K_i; see also (2.15) for the definition of DK.

We can proceed in two different ways in order to solve (2.34). We may linearize first and then reduce the result to Lagrange multipliers (Nonlinear-FETI-DP-1), or, by the help of the inverse function theorem, we can perform a nonlinear elimination and then linearize the reduced nonlinear system (Nonlinear-FETI-DP-2). Both variants will be discussed in the remainder of this thesis in Sections 2.4.1 and 2.4.3.

2.4.1 Nonlinear-FETI-DP-1 Method (Linearize First)

We now consider the first approach, denoted by Nonlinear-FETI-DP-1, in which we first linearize the system (2.34) and then reduce the resulting linear system to the space of Lagrange multipliers.

Newton linearization with respect to $(\tilde{u}, \lambda)$ of the saddle point problem (2.34) results in the linear system

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f} \\ B\tilde{u}^{(k)} \end{bmatrix}. \tag{2.36}$$

Here, $\tilde{u}^{(0)} \in \widetilde{W}$ and $\lambda^{(0)} \in V$ are initial values and our Newton iteration to solve problem (2.34) is defined as

$$\begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix}, \tag{2.37}$$

with a suitable step length $\alpha^{(k)}$. Here, as in Newton-Krylov-FETI-DP, a step length fulfilling the Wolfe conditions depicted in (2.19) and (2.20) can be used.

The linear system (2.36) can be solved as in the standard (linear) FETI-DP framework, i.e., we can eliminate the variables $\delta\tilde{u}^{(k)}$ by a step of block Gauss elimination under the assumption that $D\widetilde{K}$ is regular; see also Assumption 2 in Section 2.5.1. This yields

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ 0 & -B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} f_{NL1}(\tilde{u}^{(k)}, \lambda^{(k)}) \\ -d_{NL1}(\tilde{u}^{(k)}, \lambda^{(k)}) \end{bmatrix} \tag{2.38}$$

where

$$f_{NL1}(\tilde{u}, \lambda) = \widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f}, \tag{2.39}$$

$$d_{NL1}(\tilde{u}, \lambda) = -B\tilde{u} + B(D\widetilde{K}(\tilde{u}))^{-1}(\widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f}). \tag{2.40}$$

The second row in (2.38) yields the FETI-DP system. As in the case of standard linear FETI-DP, we have reduced the problem of computing the update in (2.37) to the solution of a linear system operating on the Lagrange multipliers, and it remains to solve

$$F_{NL1}(\tilde{u}^{(k)})\delta\lambda^{(k)} = d_{NL1}(\tilde{u}^{(k)},\lambda^{(k)}), \tag{2.41}$$

where

$$F_{NL1}(\tilde{u}) := B(D\widetilde{K}(\tilde{u})^{-1})B^T. \tag{2.42}$$

This system can be solved by a preconditioned Krylov iteration using one of the standard FETI-DP preconditioners M^{-1}, e.g., the Dirichlet preconditioner [133]. Note that here continuity of the solution may not be reached until convergence of the Newton iteration. It can easily be seen that, if the linearization is performed at the same $\tilde{u}$, then $F_{NL1}(\tilde{u}) = F_{NK}(\tilde{u})$. This does not imply that the methods are identical. For the same $\tilde{u}$, the Nonlinear-FETI-DP-1 approach differs from the standard Newton-Krylov-FETI-DP by using a different right hand side, i.e., $d_{NK}(\tilde{u}) = d_{NL1}(\tilde{u},0) - B\tilde{u}$; cf., also (2.24) and (2.36). Note that, as a result of

$$B\,\delta\tilde{u}^{(k)} = B\tilde{u}^{(k)} \tag{2.43}$$

the jumps in the Newton update will be present only if the initial value $\tilde{u}^{(0)}$ has jumps. This special jump condition (2.43) gives us the possibility of choosing initial values $\tilde{u}^{(0)}$ for Nonlinear-FETI-DP-1 with nonzero jumps in all non-primal interface variables; see Section 2.4.2. In general, this is not possible in standard Newton-Krylov methods, where a continuous initial value is necessary.

2.4.2 Choosing Initial Values for Nonlinear-FETI-DP-1

A suitable initial value $\tilde{u}^{(0)}$ for Nonlinear-FETI-DP-1 has to be continuous in all primal variables and should provide a good local approximation to the given problem. But it may be discontinuous in the dual variables. A possible choice of an initial value $\tilde{u}^{(0)}$ can be obtained from the solution of the nonlinear problem

$$\widetilde{K}(\tilde{u}^{(0)}) = \tilde{f} - B^T\lambda^{(0)} \tag{2.44}$$

by some Newton type iterations for some given initial value $\lambda^{(0)}$. Throughout this thesis we use $\lambda^{(0)} = 0$. This step can be seen as a nonlinear localization step. It can thus be performed not only in the initialization but also during the iteration. Nonlinear localization was already used in [25] and, of course, in [11, 117]. Note that in (2.44) we solve local nonlinear subdomain problems which are only coupled in the primal unknowns. This step thus requires only communication in the primal variables and is otherwise completely solvable in parallel. More sophisticated choices for $\lambda^{(0)}$, especially for hard problems as, e.g., incompressible hyperelasticity and problems with strong local nonlinearities, are also possible.

2.4.3 Nonlinear-FETI-DP-2 Method (Eliminate First)

Let us now consider the second approach denoted by Nonlinear-FETI-DP-2. Instead of linearizing the nonlinear saddle point problem (2.33), we may first perform a nonlinear elimination of the variables $\tilde{u}$. From (2.23), we recall

$$\widetilde{K}(\tilde{u}) = R_\Pi^T K(R_\Pi \tilde{u}).$$

Assuming that $\widetilde{K}(\tilde{u})$ is locally invertible, see Section 2.5.1 for a more detailed discussion, the first equation of (2.33) can be written as

$$\tilde{u} = \widetilde{K}^{-1}(\tilde{f} - B^T\lambda), \tag{2.45}$$

where $\widetilde{K}^{-1}$ is the inverse operator of $\widetilde{K}$. Inserting (2.45) into the continuity condition in (2.33), we obtain

$$F(\lambda) := B\widetilde{K}^{-1}(\tilde{f} - B^T\lambda) = 0. \tag{2.46}$$

To be able to apply the inverse function theorem, we assume that $\widetilde{K}(\cdot)$ is continuously differentiable in a neighborhood of $\tilde{u}^*$, where $(\tilde{u}^*, \lambda^*)$ is a solution of (2.33). For more details, see Assumption 2 in Section 2.5.1. Again, we use a Newton type iteration to solve (2.46), and obtain the iteration

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}(D_\lambda F(\lambda^{(k)}))^{-1} F(\lambda^{(k)}). \tag{2.47}$$

We compute $D_\lambda F(\lambda)$ using the chain rule, the inverse function theorem, and (2.45),

$$\begin{aligned} D_\lambda F(\lambda) &= D_\lambda(B\widetilde{K}^{-1}(\tilde{f} - B^T\lambda)) &&= -B(D\widetilde{K}^{-1}(\tilde{f} - B^T\lambda))B^T \\ &= -B(D\widetilde{K}(\tilde{u}))^{-1}B^T &&= -B(\underbrace{R_\Pi^T(DK(R_\Pi\tilde{u})R_\Pi}_{=D\widetilde{K}(\tilde{u})})^{-1}B^T. \end{aligned} \tag{2.48}$$

Note that, since the coarse space is included in the nonlinear operator F, and choosing different primal conditions leads to different nonlinear operators F, the choice of the coarse space not only affects the Krylov subspace iteration but also the convergence of the Newton scheme. The coarse space can thus be used to improve the convergence of the Newton iteration for nonlinear problems. Therefore, we have a nonlinear coarse problem introducing a coupling of the local nonlinear problems on the subdomains. A good coarse problem should be small but at the same time the resulting coupling should lead to a good approximation of the original global nonlinear problem. We also present some numerical results in Tables 2.3 and 2.4, which show the effect of different coarse spaces on the number of Newton and Krylov iterations. A detailed description of the associated model problem and the two different coarse spaces can be found in Section 2.7.2.

Remark that the pair $(\tilde{u},\lambda)$ in (2.48) has to satisfy equation (2.45), such that we can use the inverse function theorem. We may now define

$$F_{NL2}(\tilde{u}) := B(D\widetilde{K}(\tilde{u}))^{-1}B^T. \tag{2.49}$$

We have $F_{NL2}(\tilde{u}) = -D_\lambda F(\lambda)$ and $F(\lambda) = B\tilde{u}$, if the pair $(\tilde{u},\lambda)$ fulfills (2.45).

In each Newton step, we first have to solve the nonlinear system

$$\widetilde{K}(\tilde{u}^{(k)}) = \tilde{f} - B^T\lambda^{(k)}, \tag{2.50}$$

to compute the right hand side, e.g., by some Newton type iteration. Note that the form of the systems (2.50) and (2.44) is identical. Subsequently, we obtain our Newton update $\delta\lambda^{(k)}$ by solving

$$F_{NL2}(\tilde{u}^{(k)})\delta\lambda^{(k)} = -B\tilde{u}^{(k)} \tag{2.51}$$

or, equivalently,

$$D_\lambda F(\lambda^{(k)})\delta\lambda^{(k)} = F(\lambda^{(k)}) \tag{2.52}$$

with some Krylov subspace iteration and update

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}\delta\lambda^{(k)}$$

with a suitable step length $\alpha^{(k)}$.

Note that the system matrix in (2.51) for the Krylov iteration is the same as in NL-FETI-DP-1 and NK-FETI-DP if the linearization is performed in the same point, i.e., we have $F_{NL1}(\tilde{u}) = F_{NL2}(\tilde{u}) = F_{NK}(\tilde{u})$, where $F_{NK}(\tilde{u})$ is defined as in Section 2.2. Only the right hand side differs. To obtain the right hand side and to calculate the matrix $F_{NL2}(\tilde{u}^{(k)})$ we have to solve local nonlinear subdomain problems which are only coupled in the primal unknowns.

Nevertheless, Nonlinear-FETI-DP-2 as Nonlinear-FETI-DP-1 iterate in $\widetilde{W}$ and therefore continuity on the interface is not reached until convergence. In Fig. 2.2 we give a visual comparison of the convergence behavior of Nonlinear-

FETI-DP-2 and Newton-Krylov-FETI-DP solving our second model problem introduced in Section 2.7.2.

2.4.4 Algorithms and Cost Comparison

In this section, we give an algorithmic description of our nonlinear FETI-DP methods and of NK-FETI-DP; see Figures 2.3, 2.4, and 2.5. We also roughly discuss the computational cost of the algorithms. This will help to compare the numerical results of the different methods using our Matlab implementation in Section 2.7. We also provide a more detailed discussion on the parallel implementation of the different building blocks of our nonlinear methods in Section 3.3.
Note that the Newton direction is a descent direction if the Jacobi matrix is symmetric positive definite, which is always the case in our model problems using the p-Laplace or Laplace operator; see Section 2.7 for the model problems. The tangential matrix $DK(\cdot)$ of the p-Laplace problem is only not positive definite in constant functions. Avoiding this function, Newton direction will always be a decent direction; c.f., also Section 1.2

Typical Work and Cost

As opposed to a standard Newton-Krylov-FETI-DP approach in our nonlinear FETI-DP methods weakly coupled nonlinear local problems are solved; see (2.44) and (2.50). By increasing the local computational work we aim to reduce the amount of communication as well as the need for synchronization.

For a rough cost comparison, in order to estimate the local computational work, we will count the number of *factorizations of* $\widetilde{DK}$, denoted "*#Factor.*", and, for the global communication, we will count the number of ***Krylov iterations***, denoted "*#Krylov It.*". The factorization of $\widetilde{DK}$ also includes some communication but only in the primal variables.

In all of the three algorithms, NK-FETI-DP, Nonlinear-FETI-DP-1, and Nonlinear-FETI-DP-2, we have to solve two different types of linear systems. The first type of system is of the form

$$\widetilde{DK}(\tilde{u})\,\delta\tilde{u} = rhs_1 \tag{2.53}$$

with some right hand side rhs_1. This system appears in each Newton step in the computation of the initial value in Nonlinear-FETI-DP-1, see (2.44), and in each Newton step on the right hand side of Nonlinear-FETI-DP-2, see (2.50).

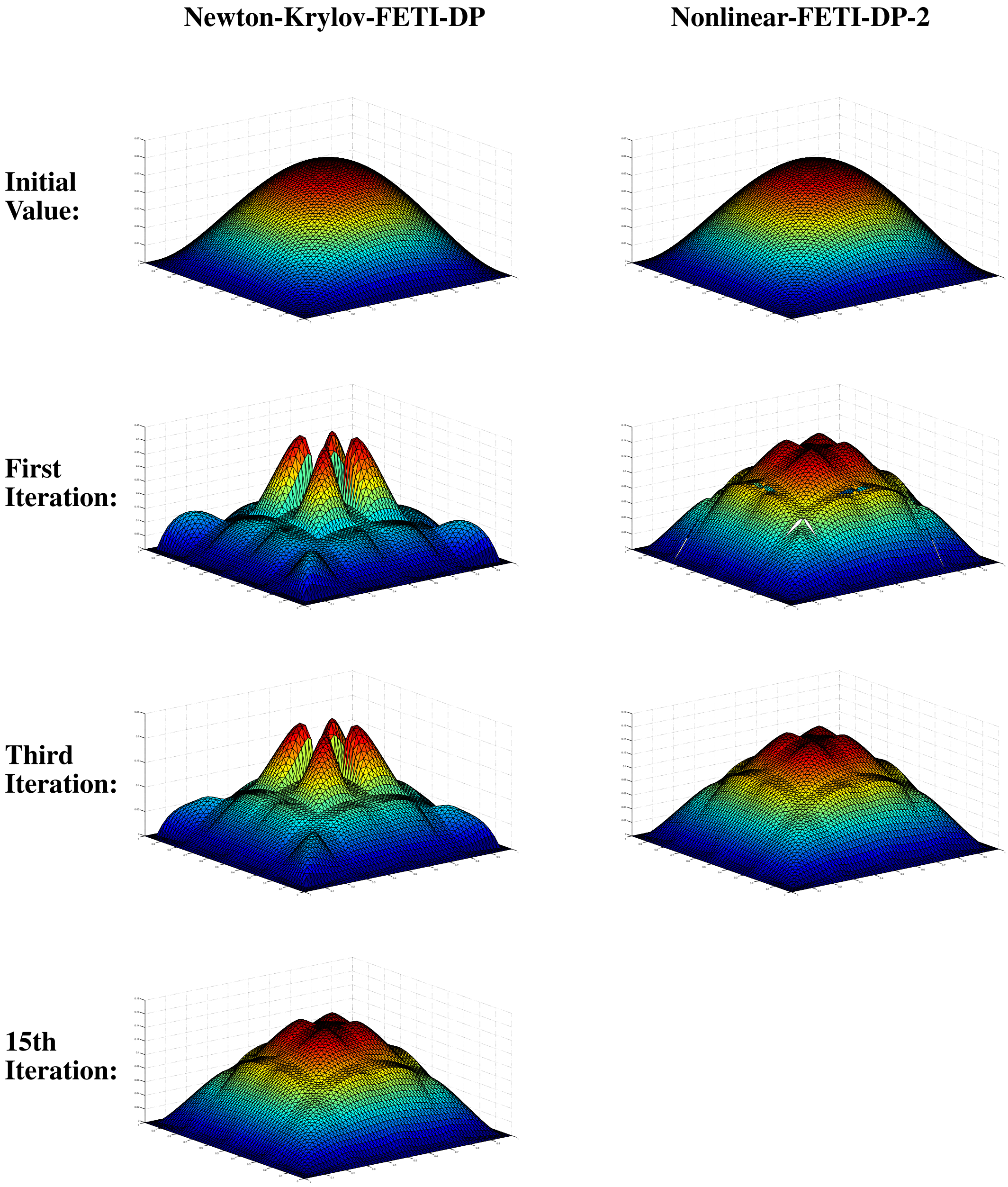

Fig. 2.2: Comparison of the convergence behavior of Newton-Krylov-FETI-DP (left column) and Nonlinear-FETI-DP-2 (right column) for our second model problem decomposed in 16 subdomains; see also Section 2.7.2 for a detailed problem description. Numerical scalability results for this problem can be found in Table 2.1. While Nonlinear-FETI-DP-2 needs three iterations until convergence, it takes Newton-Krylov-FETI-DP 15 steps to reach the same accuracy. Of course, in each Nonlinear-FETI-DP-2 iteration local nonlinear problems have to be solved to evaluate the right hand side and thus the effort of one iteration is higher.

```
NK-FETI-DP
    Init: $\tilde{u}^{(0)} \in \widehat{W}$

    for $k = 0, ..., convergence$
        build: $\widetilde{K}(\tilde{u}^{(k)})$ and $D\widetilde{K}(\tilde{u}^{(k)})$
        solve:
        $B\,(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}\,B^T\,\lambda = B\,(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}\,(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f})$        // See (2.26)
        compute:
        $\delta\tilde{u}^{(k)} = D\widetilde{K}(\tilde{u}^{(k)})^{-1}\,(\widetilde{K}(\tilde{u}^{(k)}) - \tilde{f} - B^T\lambda)$        // Compute $\delta\tilde{u}$ from $\lambda$.
        compute: steplength $\alpha^{(k)}$
        update: $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\,\delta\tilde{u}^{(k)}$

    end
```

Fig. 2.3: Algorithmic description of Newton-Krylov-FETI-DP. Figure already published in [76].

Linear systems as in (2.53) are typically solved directly in exact FETI-DP methods and thus the solution requires one *factorization of* $D\widetilde{K}$.

The second type is of the form

$$B(D\widetilde{K}(\tilde{u}))^{-1}B^T\,\delta\lambda = rhs_2 \tag{2.54}$$

with some right hand side rhs_2. This system appears on the left hand side in Nonlinear-FETI-DP-1, see (2.41), Nonlinear-FETI-DP-2, see (2.51), as well as NK-FETI-DP, see (2.26). We solve this system iteratively and matrix-free by a preconditioned Krylov method. The solution of this system requires the computation of the factorization of $D\widetilde{K}$ for the setup and, in addition, a number of Krylov iterations. In each of these iterations one forward backward substitution has to be performed.

In our cost estimate we thus consider the solution of the linear system (2.54) as requiring one *factorization of* $D\widetilde{K}$ and a certain number of *Krylov iterations*.

2.4.5 Transformation of Basis in the Nonlinear FETI-DP Methods

A well known and effective way to implement weighted or arithmetic edge averages in the FETI-DP coarse problem is the local transformation into a new basis, which can be performed independently for all edges and subdomains. Of course alternative approaches to implement coarse problems in FETI-DP methods are also possible here.

```
Nonlinear-FETI-DP-1
    Init: $\tilde{u}^{(0)} \in \widehat{W}$, $\lambda^{(0)} = 0$

    for $k = 0, ..., convergence$            // Compute initial value.
        build: $\widetilde{K}(\tilde{u}^{(k)})$ and $D\widetilde{K}(\tilde{u}^{(k)})$
        solve: $D\widetilde{K}(\tilde{u}^{(k)})\,\delta\tilde{u}^{(k)} = \widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(0)} - \tilde{f}$     // See (2.44).
        compute: steplength $\alpha^{(k)}$
        update: $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\delta\tilde{u}^{(k)}$
    end

    $\tilde{u}^{(0)} := \tilde{u}^{(k+1)}$

    for $k = 0, ..., convergence$            // Main iteration loop.
        build: $\widetilde{K}(\tilde{u}^{(k)})$ and $D\widetilde{K}(\tilde{u}^{(k)})$
        solve:
        $B\,(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T\ \delta\lambda^{(k)}$
                $= -B\tilde{u}^{(k)} + B\,(D\widetilde{K}(\tilde{u}^{(k)})^{-1}\,(\widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f}))$     // See (2.41).
        compute: $\delta\tilde{u}^{(k)} = D\widetilde{K}(\tilde{u}^{(k)})^{-1}\,(\widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - B^T\,\delta\lambda^{(k)} - \tilde{f})$
                                                        // Compute $\delta\tilde{u}$ from $\delta\lambda$.
        compute: steplength $\alpha^{(k)}$
        update: $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\,\delta\tilde{u}^{(k)}$ , $\lambda^{(k+1)} := \lambda^{(k)} - \alpha^{(k)}\,\delta\lambda^{(k)}$
    end
```

Fig. 2.4: Algorithmic description of Nonlinear-FETI-DP-1. Figure already published in [76].

Here, for robustness, we use orthogonal transformations constructed by a local Gram-Schmidt process on the edge as in [85]. In the new basis, the edge averages are represented by one nodal constraint for each former edge constraint. We mostly retain the sparsity of the tangential matrices, see [83], and the form of R_Π and B, we only have to augment R_Π with one new nodal constraint for each edge constraint and remove the new primal variables from our dual space and thus also corresponding rows from the jump matrix B. With given local and orthogonal transformations T_i, operating on W_i, we can write an $\tilde{u}_T$, where the index T marks the description in the transformed basis, in the form $\tilde{u}_T = R_{\Pi,D}^T T^T R_\Pi \tilde{u}$. Here $\tilde{u}$ is described in the nodal basis and we define the matrix $T := \mathrm{diag}(T_1, ..., T_N)$. We reformulate the saddle point problem (2.33) in the new

```
Nonlinear-FETI-DP-2
    Init: $\tilde{u}^{(0)} \in \widehat{W}$, $\lambda^{(0)} = 0$

    for $k = 0, ..., convergence$            // Main iteration loop.
        for $l = 0, ..., convergence$            // Compute right hand side.
            build: $\widetilde{K}(\tilde{u}^{(l)})$ and $D\widetilde{K}(\tilde{u}^{(l)})$
            solve: $D\widetilde{K}(\tilde{u}^{(l)})\,\delta\tilde{u}^{(l)} = \widetilde{K}(\tilde{u}^{(l)}) + B^T\lambda^{(k)} - \tilde{f}$    // See (2.50).
            compute: steplength $\alpha^{(l)}$
            update: $\tilde{u}^{(l+1)} := \tilde{u}^{(l)} - \alpha^{(l)}\delta\tilde{u}^{(l)}$
        end
        $\tilde{v}^{(k)} := \tilde{u}^{(l+1)}$
        build: $\widetilde{K}(\tilde{v}^{(k)})$ and $D\widetilde{K}(\tilde{v}^{(k)})$
        solve: $B\,(D\widetilde{K}(\tilde{v}^{(k)}))^{-1}B^T\ \delta\lambda^{(k)} = -B\,\tilde{v}^{(k)}$    // See (2.51).
        update: $\lambda^{(k+1)} := \lambda^{(k)} - \alpha^{(k)}\,\delta\lambda^{(k)}$
        $\tilde{u}^{(0)} = \tilde{v}^{(k)}$

    end
```

Fig. 2.5: Algorithmic description of Nonlinear-FETI-DP-2. Figure already published in [76].

basis by

$$\begin{aligned} R_\Pi^T T^T K(TR_\Pi \tilde{u}_T) + B^T\lambda - R_\Pi^T T^T f &= 0 \\ B\tilde{u}_T &= 0, \end{aligned} \tag{2.55}$$

or, in a shorter notation, by

$$\begin{aligned} \widetilde{K}_T(\tilde{u}_T) + B^T\lambda - R_\Pi^T T^T f &= 0 \\ B\tilde{u}_T &= 0, \end{aligned} \tag{2.56}$$

where $\widetilde{K}_T(\tilde{u}_T) := R_\Pi^T T^T K(TR_\Pi \tilde{u}_T)$.

Transformation of Basis in the Nonlinear-FETI-DP-1 Method

Let us first consider the initial value $\tilde{u}_T$ for Nonlinear-FETI-DP-1 and therefore equation (2.44). In the transformed basis, we now have to solve

$$\widetilde{K}_T(\tilde{u}_T) + B^T\lambda^{(0)} - R_\Pi^T T^T f = 0 \tag{2.57}$$

by a Newton iteration of the form

$$\tilde{u}_T^{(k+1)} = \tilde{u}_T^{(k)} - \alpha^{(k)} \delta \tilde{u}_T^{(k)}. \tag{2.58}$$

We can actually perform the complete iteration in the new basis by starting with $\tilde{u}_T^{(0)} := R_{\Pi,D}^T T^T R_\Pi \tilde{u}^{(0)}$ and with the Newton updates defined by

$$R_\Pi^T T^T DK(TR_\Pi \tilde{u}_T^{(k)})TR_\Pi \, \delta\tilde{u}_T^{(k)} = \widetilde{K}_T(\tilde{u}_T^{(k)}) + B^T \lambda^{(0)} - R_\Pi^T T^T f. \tag{2.59}$$

For a shorter notation we define $D\widetilde{K}_T(\tilde{u}_T) := R_\Pi^T T^T DK(TR_\Pi \tilde{u}_T)TR_\Pi$.
We obtain the following iteration for our Nonlinear-FETI-DP-1 method

$$\begin{bmatrix} \tilde{u}_T^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_T^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} \delta\tilde{u}_T^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix}, \tag{2.60}$$

where the update is obtained by solving

$$\begin{bmatrix} D\widetilde{K}_T(\tilde{u}_T^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_T^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{K}_T(\tilde{u}_T^{(k)}) + B^T \lambda^{(k)} - R_\Pi^T T^T f \\ B\tilde{u}_T^{(k)} \end{bmatrix}. \tag{2.61}$$

We can proceed as in the case without a transformation of basis. After convergence, we obtain the solution in the nodal basis by applying the transformation T.

In our implementation we perform all Newton iterations in the original nodal basis and only solve the linearized systems in the new basis. This is convenient, because discretized functions, gradients and tangential matrices are typically represented in the nodal basis, as well as the stopping criteria. Therefore, we remark that

$$\begin{aligned} \tilde{u}^{(k+1)} :&= R_{\Pi,D}^T TR_\Pi \tilde{u}_T^{(k+1)} \\ &= R_{\Pi,D}^T TR_\Pi \left(\tilde{u}_T^{(0)} + \sum_{i=0}^{k} \delta\tilde{u}_T^{(i)} \right) \\ &= \tilde{u}^{(0)} + \sum_{i=0}^{k} R_{\Pi,D}^T TR_\Pi \delta\tilde{u}_T^{(i)}. \end{aligned}$$

The iteration for the computation of the initial value (2.58) becomes

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} - \alpha^{(k)} R_{\Pi,D}^T TR_\Pi \delta\tilde{u}_T^{(k)} \tag{2.62}$$

where the update is obtained by solving

$$R_\Pi^T T^T DK(\tilde{u}^{(k)})\, TR_\Pi\, \delta\tilde{u}_T^{(k)} = R_\Pi^T T^T K(\tilde{u}^{(k)}) + B^T\lambda^{(0)} - R_\Pi^T T^T f.$$

Now, we will also formulate (2.60) and (2.61) in the nodal basis and obtain

$$\begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} R_{\Pi,D}^T TR_\Pi \delta\tilde{u}_T^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} \tag{2.63}$$

where the update is obtained by solving

$$\begin{bmatrix} R_\Pi^T T^T DK(\tilde{u}^{(k)})\, TR_\Pi & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_T^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} R_\Pi^T T^T K(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - R_\Pi^T T^T f \\ BR_{\Pi,D}^T T^T R_\Pi \tilde{u}^{(k)} \end{bmatrix}. \tag{2.64}$$

In this approach we have the same linearized systems because we evaluate the discrete operators in the same points, but all solutions and iterations are formulated in the nodal basis. In all cases we only have local multiplications of T or T^T with vectors $R_\Pi\tilde{u}$ or discrete operators $K(\cdot)$ and tangential matrices $DK(\cdot)$.

Transformation of Basis in the Nonlinear-FETI-DP-2 Method

Let us recall

$$\widetilde{K}_T(\tilde{u}_T) = R_\Pi^T T^T K(TR_\Pi\tilde{u}_T).$$

By performing a nonlinear elimination of $\tilde{u}_T$ in (2.55), we obtain

$$F_T(\lambda) = B\widetilde{K}_T^{-1}(R_\Pi^T T^T f - B^T\lambda)$$

and

$$D_\lambda F_T(\lambda) = -B(D\widetilde{K}_T(\tilde{u}_T))^{-1}B^T.$$

Here, the pair $(\lambda, \tilde{u}_T)$ has to satisfy the equation

$$\widetilde{K}_T(\tilde{u}_T) + B^T\lambda - R_\Pi^T T^T f = 0. \tag{2.65}$$

Now we have all ingredients to formulate Nonlinear-FETI-DP-2 completely in the new basis. The computation of a $\tilde{u}_T$, which satisfies (2.65) for the iterate $\lambda^{(k)}$, can be carried out analogously to (2.58) with $\lambda^{(k)}$ replacing $\lambda^{(0)}$ in (2.59). For the dual Newton iteration, we have

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}\delta\lambda^{(k)}$$

and the update is obtained by solving

$$B(D\widetilde{K}_T(\tilde{u}_T))^{-1}B^T\delta\lambda^{(k)} = -B\tilde{u}_T.$$

As in Nonlinear-FETI-DP-1 we can perform all iterations in the nodal basis. Therefore, we compute a variable $\tilde{u}$ such that $R_{\Pi,D}^T T^T R_\Pi \tilde{u}$ satisfies (2.65) for the iterate $\lambda^{(k)}$ analogously to (2.62). The dual update $\delta\lambda^{(k)}$ can be obtained by solving

$$B(R_\Pi^T T^T DK(\tilde{u})TR_\Pi)^{-1}B^T\delta\lambda^{(k)} = -BR_{\Pi,D}^T T^T R_\Pi \tilde{u}.$$

Again, we only perform local multiplications with T or T^T.

2.5 On the Convergence of Nonlinear FETI-DP

In general, we expect a quadratic convergence rate of Newton's method in a sufficiently small environment of the solution. It is well known that this environment can be enlarged by using a step length or trust region globalization technique; see Sections 1.3 and 1.4, and [112]. As already mentioned, we often use step lengths fulfilling the Wolfe conditions. Additionally, we describe an alternative approach to compute a step length α for Nonlinear-FETI-DP-2, which has less computational intensity; see Section 2.5.3. In Section 2.5.1, we also make some additional Assumptions which are necessary for equations (2.38) and (2.45) and thus for the derivation of our nonlinear methods. We also discuss the invertibility of $D\widetilde{K}(\cdot)$. Finally, we provide some remarks on the usage of redundant Lagrange multiplier variables in nonlinear FETI-DP methods; see Section 2.5.2.

2.5.1 Additional Assumptions

First, let us note that our nonlinear FETI-DP methods are identical to standard FETI-DP methods if they are applied to linear problems. Moreover, nonlinear FETI-DP methods using a transformation of basis are also identical to standard FETI-DP methods using a transformation of basis if applied to linear problems.

Second, as in the ASPIN method [22], the equivalence of the original problem (2.27) to the nonlinear FETI-DP formulations (2.33) and (2.46) relies on certain assumptions which we discuss briefly now.

To ensure the equivalence of the original problem (2.27) and the saddle point system (2.32), Assumption 1 has to be satisfied. From the saddle point system (2.32) we will obtain the saddle point system (2.33) by replacing some of the dual constraints by primal constraints implemented by partial assembly. This does not essentially change the solution. Let us now assume that (2.33) has a solution $(\tilde{u}^*, \lambda^*)$. Then, from the first set of equations of (2.33), we have $\widetilde{K}(\tilde{u}^*) = \tilde{f} - B^T\lambda^*$. We make the following assumption.

Assumption 2 *Let U be an open neighborhood of $\tilde{u}^*$. The function $\widetilde{K}$ is continuously differentiable in U. The derivative $D\widetilde{K}(\tilde{u}^*)$ of $\widetilde{K}$ in $\tilde{u}^*$ is a regular matrix.*

Under Assumption 2 and using the inverse function theorem, there exist some neighborhoods $U_a \subset U$ with $\tilde{u}^* \in U_a$ and V_a with $\tilde{f} - B^T\lambda^* \in V_a$ such that $\widetilde{K} : U_a \to V_a$ is a diffeomorphism. Under these assumptions, $\tilde{u}^*$ is a point of attraction of the Newton method for $\widetilde{K}(\tilde{u}^*) + B^T\lambda^* - \tilde{f} = 0$; see, e.g., Ortega and Rheinboldt [116, Section 10.2.2]. An analogous result holds for the Nonlinear-FETI-DP-1 system (2.34) and the Nonlinear-FETI-DP-2 system (2.46) if, in addition, the saddle point system on the left hand side of (2.36) is nonsingular for $(\tilde{u}^*, \lambda^*)$; this is satisfied if B has full rank. In general, B has only full rank, if we consider nonredundant Lagrange multipliers, i.e., in two dimensions, one multiplier per degree of freedom on edges, two multipliers in degrees of freedom with three neighboring subdomains, and three multipliers in degrees of freedom with four neighboring subdomains.

2.5.2 Nonlinear FETI-DP using Redundant Lagrange Multipliers

In linear FETI-DP methods the usage of redundant Lagrange multipliers is common and the Krylov subspace method converges to the correct solution since it operates on range(B). The same holds for Newton-Krylov-FETI-DP, since the Lagrange multipliers only exist in the linearization and, as a result of that, do not affect the Newton iteration. In nonlinear FETI-DP methods, the Newton iteration directly acts on the Lagrange multiplier variables and also affects the right hand sides of the linearized systems.

Let us recall the nonlinear FETI-DP master system from equation (2.34) in the exact solution $[\tilde{u}^{*^T}, \lambda^{*^T}]^T$

$$\begin{aligned} \widetilde{K}(\tilde{u}^*) + B^T\lambda^* - \tilde{f} &= 0 \\ B\tilde{u}^* &= 0. \end{aligned} \tag{2.66}$$

Assuming that the jump matrix B corresponds to a set of redundant Lagrange multiplier variables enforcing continuity on the interface, we can formulate an equivalent FETI-DP master system using a maximal subset of nonredundant Lagrange multipliers and the corresponding jump matrix B_1 in the exact solution

$[\tilde{u}_1^{*^T}, \lambda_1^{*^T}]^T$

$$\begin{aligned} \widetilde{K}(\tilde{u}_1^*) + B_1^T \lambda_1^* - \tilde{f} &= 0 \\ B_1 \tilde{u}_1^* &= 0. \end{aligned} \tag{2.67}$$

Obviously, we have $\tilde{u}^* = \tilde{u}_1^*$, since B_1 and B enforce the same set of constraints. In the remainder of this section, we will show that nonlinear FETI-DP applied to the nonlinear master system in (2.66) converges to the same result as nonlinear FETI-DP applied to (2.67), and both methods have the same convergence rate. For simplicity we assume the decomposition

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}. \tag{2.68}$$

Since all n_2 rows of B_2 are linear combinations of the n_1 rows of B_1, we can represent B_2 by $B_2 = L_1 B_1$, where $L_1 \in \mathbb{R}^{n_2 \times n_1}$. We also have

$$B = L \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \tag{2.69}$$

with

$$L := \begin{bmatrix} I_{n_1} & 0 \\ L_1 & I_{n_2} \end{bmatrix}$$

is a regular matrix. Here, I_{n_1} and I_{n_2} are identity matrices of matching sizes.

Nonlinear-FETI-DP-1 with Redundant Lagrange Multipliers

We first consider Nonlinear-FETI-DP-1. Let us recall iteration (2.37) to solve (2.66):

$$\begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix}. \tag{2.70}$$

The update is obtained by solving the linear system

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}^{(k)}) + B^T \lambda^{(k)} - \tilde{f} \\ B\tilde{u}^{(k)} \end{bmatrix}. \tag{2.71}$$

Let us assume we start within the convergence radius of Newton's method. Since we first reduce system (2.71) to the Lagrange multipliers and solve the resulting positive semidefinite system with a Krylov subspace method, we obtain a valid update $\delta\lambda^{(k)}$. Under Assumption 2, $D\widetilde{K}(\tilde{u}^{(k)})$ is regular and thus we get an update $\delta\tilde{u}^{(k)}$, which is unique with respect to $\delta\lambda^{(k)}$. Analogously, the Nonlinear-FETI-DP-1 iteration in order to solve (2.67) reads

$$\begin{bmatrix} \tilde{u}_1^{(k+1)} \\ \lambda_1^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_1^{(k)} \\ \lambda_1^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} \delta\tilde{u}_1^{(k)} \\ \delta\lambda_1^{(k)} \end{bmatrix}, \tag{2.72}$$

with updates defined by

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}_1^{(k)}) & B_1^T \\ B_1 & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_1^{(k)} \\ \delta\lambda_1^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}_1^{(k)}) + B_1^T\lambda_1^{(k)} - \tilde{f} \\ B_1\tilde{u}_1^{(k)} \end{bmatrix}. \tag{2.73}$$

The jump matrix B_1 has full row rank and, under Assumption 2, $D\widetilde{K}(\tilde{u}_1^{(k)})$ is invertible. The saddle point matrix in equation (2.73) is therefore regular and has a unique solution $(\delta\tilde{u}_1^{(k)}, \delta\lambda_1^{(k)})$. If we start in a sufficiently small neighborhood of the solution $(\tilde{u}_1^*, \lambda_1^*)$ of equation (2.67) , we have $(\tilde{u}_1^{(k)}, \lambda_1^{(k)}) \to (\tilde{u}_1^*, \lambda_1^*)$ with a quadratic convergence rate. In the remainder of this section, we will show that (2.70) and (2.72) have the same convergence behavior in $\widetilde{W}$.

Remark 1 *Let $(\tilde{u}^{(0)}, \lambda^{(0)})$ be the initial value of Newton's iteration defined by (*2.70*) and (*2.71*) and let $(\tilde{u}_1^{(0)}, \lambda_1^{(0)})$ be the initial value of Newton's iteration defined by (*2.72*) and (*2.73*). Let us assume that for the chosen initial value $(\tilde{u}_1^{(0)}, \lambda_1^{(0)})$ the iteration (*2.72*) converges to the unique solution $(\tilde{u}_1^*, \lambda_1^*)$. If $\tilde{u}_1^{(0)} = \tilde{u}^{(0)}$ and*

$$\lambda_1^{(0)} = R_{n_1} L^T \lambda^{(0)}, \tag{2.74}$$

where $R_{n_1} = \begin{bmatrix} I_{n_1} & 0 \end{bmatrix}$ restricts a vector of length $(n_1 + n_2)$ to the first n_1 entries, the following two equations will hold in all iterations:

$$\delta\tilde{u}_1^{(k)} = \delta\tilde{u}^{(k)} \tag{2.75}$$

and

$$\delta\lambda_1^{(k)} = R_{n_1} L^T \delta\lambda^{(k)}. \tag{2.76}$$

If $\tilde{u}_1^{(k+1)}$ satisfies a chosen stopping criterion, $\tilde{u}^{(k+1)}$ fulfill the same criterion and both iterations have the same convergence behavior in $\widetilde{W}$.

Proof. Let us note that assumption (2.74) is always fulfilled for $\lambda^{(0)} := (\lambda_1^{(0)^T}, 0)^T$. Next, we assume that Remark 1 holds for all Newton steps $0 \le i < k$, and thus we have $\delta\tilde{u}_1^{(i)} = \delta\tilde{u}^{(i)}$ and $\delta\lambda_1^{(i)} = R_{n_1} L^T \delta\lambda^{(i)}$ for all $0 \le i < k$. We then obtain

$$\tilde{u}^{(k)} = \tilde{u}^{(0)} + \sum_{i=0}^{k-1} \delta\tilde{u}^{(i)} = \tilde{u}_1^{(0)} + \sum_{i=0}^{k-1} \delta\tilde{u}_1^{(i)} = \tilde{u}_1^{(k)}. \tag{2.77}$$

Since the solution $(\delta\tilde{u}_1^{(k)}, \delta\lambda_1^{(k)})$ of system (2.73) is unique, it is sufficient to show that $(\delta\tilde{u}^{(k)}, R_{n_1} L^T \delta\lambda^{(k)})$ solves system (2.73), which immediately proves equalities (2.76) and (2.75). For the first row of (2.73) we have to show

$$D\widetilde{K}(\tilde{u}_1^{(k)})\delta\tilde{u}^{(k)} + B_1^T R_{n_1} L^T \delta\lambda^{(k)} = \widetilde{K}(\tilde{u}_1^{(k)}) + B_1^T \lambda_1^{(k)} - \tilde{f}.$$

Therefore, we have

$$\begin{aligned}
& D\widetilde{K}(\tilde{u}_1^{(k)})\delta\tilde{u}^{(k)} + B_1^T R_{n_1} L^T \delta\lambda^{(k)} \\
\overset{(2.77)}{=} \quad & D\widetilde{K}(\tilde{u}^{(k)})\delta\tilde{u}^{(k)} + \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T \delta\lambda^{(k)} \\
= \quad & D\widetilde{K}(\tilde{u}^{(k)})\delta\tilde{u}^{(k)} + B^T \delta\lambda^{(k)} \\
\overset{(2.71)}{=} \quad & \widetilde{K}(\tilde{u}^{(k)}) + B^T \lambda^{(k)} - \tilde{f} \\
= \quad & \widetilde{K}(\tilde{u}^{(k)}) + \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T \lambda^{(k)} - \tilde{f} \\
= \quad & \widetilde{K}(\tilde{u}^{(k)}) + \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T \left(\lambda^{(0)} + \sum_{i=0}^{k-1} \delta\lambda^{(i)} \right) - \tilde{f} \\
= \quad & \widetilde{K}(\tilde{u}^{(k)}) + \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T \lambda^{(0)} + \sum_{i=0}^{k-1} \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T \delta\lambda^{(i)} - \tilde{f} \\
= \quad & \widetilde{K}(\tilde{u}^{(k)}) + B_1^T \lambda_1^{(0)} + \sum_{i=0}^{k-1} B_1^T \delta\lambda_1^{(i)} - \tilde{f} \\
= \quad & \widetilde{K}(\tilde{u}_1^{(k)}) + B_1^T \lambda_1^{(k)} - \tilde{f}.
\end{aligned}$$

It remains to prove that $(\delta\tilde{u}^{(k)}, R_{n_1} L^T \delta\lambda^{(k)})$ is also a solution for the second row of (2.73):

$$\begin{aligned}
& B\delta\tilde{u}^{(k)} = B\tilde{u}^{(k)} \\
\Leftrightarrow \quad & L \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \delta\tilde{u}^{(k)} = L \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tilde{u}^{(k)} \\
\Leftrightarrow \quad & \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \delta\tilde{u}^{(k)} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tilde{u}^{(k)} \\
\Leftrightarrow \quad & B_1 \delta\tilde{u}^{(k)} = B_1 \tilde{u}^{(k)} \\
\overset{(2.77)}{\Leftrightarrow} \quad & B_1 \delta\tilde{u}^{(k)} = B_1 \tilde{u}_1^{(k)}.
\end{aligned}$$

By adding the updates, we directly obtain $\tilde{u}_1^{(k+1)} = \tilde{u}^{(k+1)}$ and $\lambda_1^{(k+1)} = R_{n_1} L^T \lambda^{(k+1)}$. The initial step of this induction can be shown analogously by choosing $k = 0$. □

Nonlinear-FETI-DP-2 with Redundant Lagrange Multipliers

We will now show the same independence of the choice of Lagrange multipliers for Nonlinear-FETI-DP-2. In Nonlinear-FETI-DP-2 we get the update of the Lagrange multipliers $\delta\lambda^{(k)}$ by solving

$$B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1} B^T \delta\lambda^{(k)} = -B\tilde{u}^{(k)}, \tag{2.78}$$

where $\tilde{u}^{(k)}$ has to fulfill the equation

$$\widetilde{K}(\tilde{u}^{(k)}) = \tilde{f} - B^T \lambda^{(k)}, \tag{2.79}$$

which can be solved by some Newton iteration. Again, with (2.69), we can formulate a similar iteration considering nonredundant Lagrange multipliers. We obtain

$$B_1(D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1} B_1^T \delta\lambda_1^{(k)} = -B\tilde{u}_1^{(k)}, \tag{2.80}$$

where $\tilde{u}_1^{(k)}$ has to fulfill the equation

$$\widetilde{K}(\tilde{u}_1^{(k)}) = \tilde{f} - B_1^T \lambda_1^{(k)}. \tag{2.81}$$

This leads to the following remark analogously to Remark 1.

Remark 2 *Let $\lambda^{(0)}$ be the initial value of a Newton iteration with updates defined by (*2.78*) and let $\lambda_1^{(0)}$ be the initial value of a Newton iteration with updates defined by (*2.80*). Let the iteration with updates defined by (*2.80*) converge for the chosen initial value $\lambda_1^{(0)}$ to the unique solution λ_1^*. The corresponding unique solution $\tilde{u}^*$ is obtained by solving (*2.81*). Given that*

$$\lambda_1^{(0)} = R_{n_1} L^T \lambda^{(0)},$$

with $R_{n_1} = \begin{bmatrix} I_{n_1} & 0 \end{bmatrix}$ restricts a vector of length n to the first n_1 entries, the following equations hold in all Newton steps:

$$\delta\tilde{u}_1^{(k)} = \delta\tilde{u}^{(k)} \tag{2.82}$$

and

$$\delta\lambda_1^{(k)} = R_{n_1} L^T \delta\lambda^{(k)}. \tag{2.83}$$

If $\tilde{u}_1^{(k+1)}$ satisfies a chosen stopping criterion, $\tilde{u}^{(k+1)}$ fulfills the same criterion and both iterations have the same convergence behavior in $\widetilde{W}$.

Proof. Since the matrix B_1 has full row rank and $D\widetilde{K}(\tilde{u}^{(k)})$ is positive definite, $B_1(D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}B_1^T$ is regular and we have a unique solution $\delta\lambda_1^{(k)}$. We first show $\delta\lambda_1^{(k)} = R_{n_1}L^T\delta\lambda^{(k)}$ under the assumption $\tilde{u}_1^{(k)} = \tilde{u}^{(k)}$. Because of the uniqueness of $\delta\lambda_1^{(k)}$ it is sufficient to show that $R_{n_1}L^T\delta\lambda^{(k)}$ solves (2.80). Therefore, we show the equivalence

$$\begin{aligned} & & B_1(D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}B_1^TR_{n_1}L^T\delta\lambda^{(k)} &= -B_1\tilde{u}_1^{(k)} \\ \Leftrightarrow & & B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T\delta\lambda^{(k)} &= -B\tilde{u}^{(k)}. \end{aligned}$$

Therefore, we have

$$\begin{aligned} & & B_1(D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}B_1^TR_{n_1}L^T\delta\lambda^{(k)} &= -B_1\tilde{u}_1^{(k)} \\ \Leftrightarrow & & B_1(D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}\begin{bmatrix} B_1^T & 0 \end{bmatrix}L^T\delta\lambda^{(k)} &= -B_1\tilde{u}^{(k)} \\ \Leftrightarrow & & B_1(D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}B^T\delta\lambda^{(k)} &= -B_1\tilde{u}^{(k)} \\ \Leftrightarrow & & B_1((D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}B^T\delta\lambda^{(k)} + \tilde{u}^{(k)}) &= 0 \\ \Leftrightarrow & & L\begin{bmatrix} B_1 \\ 0 \end{bmatrix}((D\widetilde{K}(\tilde{u}_1^{(k)}))^{-1}B^T\delta\lambda^{(k)} + \tilde{u}^{(k)}) &= 0 \\ \Leftrightarrow & & B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T\delta\lambda^{(k)} &= -B\tilde{u}^{(k)}. \end{aligned}$$

We proved that for a solution $\delta\lambda^{(k)}$ of (2.78), and under the assumption $\tilde{u}^{(k)} = \tilde{u}_1^{(k)}$, $R_{n_1}L^T\delta\lambda^{(k)}$ is the unique solution of (2.80). Therefore, we obtain $\delta\lambda_1^{(k)} = R_{n_1}L^T\delta\lambda^{(k)}$. It is left to verify the correctness of the assumption $\tilde{u}_1^{(k)} = \tilde{u}^{(k)}$. Therefore, it is sufficient to show the equality of the right hand sides of

(2.79) and (2.81) assuming that $\delta\lambda_1^{(i)} = R_{n_1}L^T\delta\lambda^{(i)}$, $\forall 0 \leq i < k$. We have:

$$\begin{aligned}
& \tilde{f} - B^T\lambda^{(k)} \\
= \quad & \tilde{f} - \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T\lambda^{(0)} - \sum_{i=0}^{k-1} \begin{bmatrix} B_1^T & 0 \end{bmatrix} L^T\delta\lambda^{(i)} \\
= \quad & \tilde{f} - B_1^T R_{n_1} L^T\lambda^{(0)} - \sum_{i=0}^{k-1} B_1^T R_{n_1} L^T\delta\lambda^{(i)} \\
= \quad & \tilde{f} - B_1^T\lambda_1^{(0)} - \sum_{i=0}^{k-1} B_1^T\delta\lambda_1^{(i)} \\
= \quad & \tilde{f} - B_1^T\lambda_k^{(k)}.
\end{aligned}$$

We easily get the initial step of the induction by repeating this proof with $k = 0$. □

Summarizing, we have shown that the convergence of the Newton iteration in nonlinear FETI-DP methods is independent of the choice of the jump matrix B. Therefore, all common FETI-DP jump matrices can be used in our nonlinear methods. Of course, as in linear FETI-DP, the choice of the jump conditions may slightly affect the convergence of the Krylov method.
Let us briefly remark that in the case of redundant Lagrange multipliers updates $\delta\lambda^{(k)}$ in nonlinear FETI-DP methods may be of the form $\delta\lambda^{(k)} = \delta\tilde{\lambda}^{(k)} + \delta\hat{\lambda}^{(k)}$ with $\delta\hat{\lambda}^{(k)} \in$ ker (B^T) and $\delta\hat{\lambda}^{(k)} \neq 0$. As we have shown in this section, this will not affect the convergence of the method. But it might influence the stopping criterion. For example, a stopping criterion including the condition $||\delta\lambda|| \leq \varepsilon$ may never be fulfilled. We suggest to use stopping criteria considering the residual instead of the Newton update. This would include $B^T\delta\lambda$ instead of $\delta\lambda$. Also a criterion considering $\delta\tilde{u}$ and $B\tilde{u}$ is possible.

2.5.3 Improving the Convergence of Nonlinear-FETI-DP-2

Since we generate a fully nonlinear problem $F(\lambda^*) = 0$ in the dual variables by nonlinear elimination of the variables $\tilde{u}$, we will collect some remarks on the convergence of the resulting Newton iteration (2.47). We can hope for quadratic convergence of the Newton method if we start in a sufficiently small neighborhood of the solution λ^*. In general, e.g., in nonlinear continuum mechanics, sufficient conditions for local quadratic convergence of the Newton method may not hold. The radius of convergence can be enlarged if a suitable globalization strategy is used. Here, we only consider line search methods. Trust region methods can also be used for the subdomain problems as well as for the coarse problem. In nonlinear multilevel methods for elasticity, globalization using recursive trust region methods has been a successful approach; see [63].

Choosing the Initial Value and Choice of the Coarse Space

For an initial value $\lambda^{(0)}$ we have

$$F(\lambda^{(0)}) = B\tilde{u}$$

for a $\tilde{u}$ which satisfies the equation

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

Thus, we may search a $\lambda^{(0)}$ such that the jump on the interface of the corresponding $\tilde{u}$ is sufficiently small. Alternatively, for a given $\lambda^{(0)}$, we can manipulate our primal constraints, such that a $\tilde{u}$ with sufficiently small jumps in the dual variables, satisfies

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

Recently developed strategies of adaptive coarse space selection for standard FETI-DP and BDDC methods, see, e.g., [38,39,82,106,131], may be especially valuable in this nonlinear context since they can help to improve the convergence of the Newton method. Also a coarse space based on a zero-net flux condition, as suggested in [57, 59], may improve convergence for almost incompressible elasticity problems.

Approach for a Step Length Calculation for the Nonlinear-FETI-DP-2 Method

In our Newton iteration

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}(D_\lambda F(\lambda^{(k)}))^{-1}F(\lambda^{(k)}),$$

we still need to define a suitable step length. In some of our numerical experiments a good step length is necessary for convergence, especially if the FETI-DP coarse space is insufficient. In this special case, for Armijo or Wolfe conditions, the expression $F(\lambda^{(k)} - \alpha\delta\lambda^{(k)})$ has to be computed for several α. In the context of Nonlinear-FETI-DP-2, for each α, a weakly coupled nonlinear system

$$F(\lambda^{(k)} - \alpha\delta\lambda^{(k)}) = B\tilde{u} = B(\widetilde{K}^{-1}(\tilde{f} - B^T(\lambda^{(k)} - \alpha\delta\lambda^{(k)}))) = 0 \tag{2.84}$$

has to be solved, e.g., using a Newton iteration. We note that the computational work is mostly local. At this point we should remember that we have to compute

$$\tilde{u} = \widetilde{K}^{-1}(\tilde{f} - B^T \lambda^{(k+1)}) \tag{2.85}$$

in every step of our algorithm, in order to compute the right hand side; see (2.50).

Here, we describe our strategy to save computational cost. First, it is often not necessary to solve (2.84) until convergence. Second, let us assume a backtracking strategy (see also 1.1) for α, e.g., $\alpha^{(0)} := 1$ and then $\alpha^{(i)} = \rho^i \alpha^{(0)}$, $i = 1, 2, \ldots$, where $0 < \rho < 1$. In this case, $\tilde{u}$ resulting from (2.84) can be reused as a start value for the iteration for the next value of α.

Note that, in our FETI-DP method, we want to minimize the jump, e.g., the norm $||B\tilde{u}|| = ||F(\lambda^{(k)} - \alpha\delta\lambda^{(k)})||$. We thus solve equation (2.84) until, e.g., stagnation of $||B\tilde{u}||$.

Finally the value of $\tilde{u}$ resulting from this minimization can also be reused when computing (2.85). Once, we have found the final step length α, we choose the stopping criterion defined in Section 2.7.2.

In cases where our simple and less expensive step length strategy is not sufficient for convergence, we suggest to verify the Wolfe condition, cf. (2.19) and (2.20), or even the strong Wolfe condition; see [112]. This can be combined with the computation of $\tilde{u}$ in (2.85) in a similar way, but will be more expensive.

2.6 Nonlinear BDDC Method

In the spirit of our nonlinear FETI-DP methods we can also formulate a nonlinear BDDC method. A nonlinear Neumann-Neumann or BDD method was already introduced by Bordeu et al. in [11]. In our nonlinear BDDC method we will eliminate all interior variables and build a nonlinear Schur complement, which then can be linearized. This approach is thus similar to Nonlinear-FETI-DP-2.

2.6.1 Newton-Krylov-BDDC

We briefly review the classical use of the linear BDDC preconditioner within a Newton-Krylov approach. Here, we recall the fully assembled system (2.16) operating on V^h using the nonoverlapping domain decomposition of Ω into subdomains Ω_i , $i = 1, ..., N$, which reads

$$R^T K(Ru) - R^T f = 0, \tag{2.86}$$

where K is defined in (2.17). This system is solved by a Newton iteration

$$u^{(k+1)} = u^{(k)} - \alpha^{(k)} \delta u^{(k)}$$

with the update $\delta u^{(k)}$ obtained by

$$R^T DK(Ru^{(k)}) R\delta u^{(k)} = R^T K(Ru^{(k)}) - R^T f. \tag{2.87}$$

As usual, we partition the variables into interior variables denoted by u_I and interface variables denoted by u_Γ. This leads us to the following partition of the tangential matrix

$$DK(Ru^{(k)}) = \begin{bmatrix} DK(Ru^{(k)})_{II} & DK(Ru^{(k)})_{I\Gamma} \\ DK(Ru^{(k)})_{\Gamma I} & DK(Ru^{(k)})_{\Gamma\Gamma} \end{bmatrix}$$

and also the partition of the right hand side

$$K(Ru^{(k)}) - f = \begin{bmatrix} (K(Ru^{(k)}) - f)_I \\ (K(Ru^{(k)}) - f)_\Gamma \end{bmatrix} = \begin{bmatrix} K(Ru^{(k)})_I - f_I \\ K(Ru^{(k)})_\Gamma - f_\Gamma \end{bmatrix}.$$

Here, I denotes the set of interior variables and Γ the set of interface variables. Using the BDDC algorithm to solve the linearized system (2.87) we first eliminate the interior variables and solve for the assembled interface variables

$$M_{BDDC}^{-1}(Ru^{(k)})\; S_g(Ru^{(k)})\; \delta u_g^{(k)} = M_{BDDC}^{-1}(Ru^{(k)})\; g_g(Ru^{(k)})$$

by some Krylov iteration. Here, we have the assembled Schur complement $S_g(u)$ defined by

$$S_g(u) := R_\Gamma^T S_\Gamma(u) R_\Gamma = R_\Gamma^T (DK(u)_{\Gamma\Gamma} - DK(u)_{\Gamma I} (DK(u)_{II})^{-1} DK(u)_{I\Gamma}) R_\Gamma, \tag{2.88}$$

with the restriction operator $R_\Gamma^T := \left[R_\Gamma^{(1)T}, ..., R_\Gamma^{(N)T} \right]^T$, where $R_\Gamma^{(i)}$ is the restriction from the global interface Γ to the local interface on subdomain Ω_i. The BDDC preconditioner is defined by

$$M_{BDDC}^{-1}(u) := R_D^T (R_\Pi^T DK(u) R_\Pi)^{-1} R_D \tag{2.89}$$

with

$$R_D^T := \begin{bmatrix} R_{\Delta,D}^T R_\Delta^B & 0 \\ 0 & I_\Pi \end{bmatrix},$$

and $R_{\Delta,D}^T R_\Delta^B$ is the weighted restriction from the set $[I,\Delta]$ to Δ. The right hand side g_g is defined by

$$g_g(u) := R_\Gamma^T((K(u)-f)_\Gamma - DK(u)_{\Gamma I}(DK(u)_{II})^{-1}(K(u)-f)_I).$$

Finally, we get the full Newton update by

$$\delta u^{(k)} := \begin{bmatrix} (DK(Ru^{(k)})_{II})^{-1}((K(Ru^{(k)})-f)_I - DK(Ru^{(k)})_{I\Gamma}\delta u_g^{(k)}) \\ \delta u_g^{(k)} \end{bmatrix}.$$

2.6.2 Nonlinear Schur Complement

To formulate a nonlinear BDDC algorithm let us first introduce the nonlinear Schur complement system. For the moment, let us assume that $K(u) = f$ is the discretization of our nonlinear problem on a single domain, e.g., this could be a subdomain. Let us recall the partition of all degrees of freedom into the set of interface variables Γ and interior variables I from the last section. This leads us to the following partition of the operator $K(u)$ and the right hand side f

$$K(u) - f = \begin{bmatrix} K_I(u_I,u_\Gamma) \\ K_\Gamma(u_I,u_\Gamma) \end{bmatrix} - \begin{bmatrix} f_I \\ f_\Gamma \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.90}$$

We also partition the tangential matrix $DK(u)$ in the same way and obtain

$$DK(u) = \begin{bmatrix} D_{u_I}K_I(u_I,u_\Gamma) & D_{u_\Gamma}K_I(u_I,u_\Gamma) \\ D_{u_I}K_\Gamma(u_I,u_\Gamma) & D_{u_\Gamma}K_\Gamma(u_I,u_\Gamma) \end{bmatrix} = \begin{bmatrix} DK(u_I,u_\Gamma)_{II} & DK(u_I,u_\Gamma)_{I\Gamma} \\ DK(u_I,u_\Gamma)_{\Gamma I} & DK(u_I,u_\Gamma)_{\Gamma\Gamma} \end{bmatrix}. \tag{2.91}$$

Under the sufficient assumptions that K_I is continuously differentiable, that there exists an (u_I,u_Γ), which satisfies $K_I(u_I,u_\Gamma) - f_I = 0$, and that $D_{u_I}K_I(u_I,u_\Gamma)$ is invertible, there exists an implicit function $h(u_\Gamma) = u_I$ and in a neighborhood of u_Γ we have

$$0 = K_I(u_I,u_\Gamma) - f_I = K_I(h(u_\Gamma),u_\Gamma) - f_I.$$

From the implicit function theorem we then obtain the derivative of h,

$$Dh(u_\Gamma) = -(DK(h(u_\Gamma),u_\Gamma)_{II})^{-1}DK(h(u_\Gamma),u_\Gamma)_{\Gamma\Gamma}. \tag{2.92}$$

Now, we can define the nonlinear Schur complement by

$$S(u_\Gamma) := K_\Gamma(h(u_\Gamma),u_\Gamma) - f_\Gamma \tag{2.93}$$

and, using the chain rule in combination with (2.92), the derivative of S is given by

$$\begin{aligned} DS(u_\Gamma) &= D_{u_I}K_\Gamma(h(u_\Gamma),u_\Gamma)Dh(u_\Gamma)+D_{u_\Gamma}K_\Gamma(h(u_\Gamma),u_\Gamma) \\ &= -DK(h(u_\Gamma),u_\Gamma)_{\Gamma I}\,(DK(h(u_\Gamma),u_\Gamma)_{II})^{-1}\,DK(h(u_\Gamma),u_\Gamma)_{I\Gamma}\,+\,DK(h(u_\Gamma),u_\Gamma)_{\Gamma\Gamma}. \end{aligned}$$

Let us note that the derivative of the nonlinear Schur complement DS is the same as the Schur complement S_Γ of the derivative of the nonlinear operator K; see also (2.88) in Section 2.6.1.

If we now consider more than one subdomain, then $S(u_\Gamma)$ has a block form. Let us reintroduce the index $i=1,...,N$ and define the nonlinear Schur complement on a subdomain Ω_i by $S_i(u_{i_\Gamma})$. Then, we have

$$S(u_\Gamma) := \left[S_1(u_{1_\Gamma})^T,...,S_N(u_{N_\Gamma})^T\right]^T, \tag{2.94}$$

where $u_\Gamma := \left[u_{1_\Gamma}^T,...,u_{N_\Gamma}^T\right]^T$.

2.6.3 Nonlinear BDDC Formulation

Our nonlinear BDDC method is obtained by linearizing the nonlinear Schur complement and using a BDDC preconditioner to solve the linearized problem. As in the Nonlinear-FETI-DP-2 method we have to solve local nonlinear problems on the right hand side. We recall the assembly operator on the interface $R_\Gamma^T := \left[R_\Gamma^{(1)T},...,R_\Gamma^{(N)T}\right]^T$. Next, we solve the assembled nonlinear Schur complement system, cf. (2.94),

$$R_\Gamma^T S(R_\Gamma \bar{u}_\Gamma) = 0$$

by a Newton type iteration of the form

$$\bar{u}_\Gamma^{(k+1)} = \bar{u}_\Gamma^{(k)} - \alpha^{(k)}\delta\bar{u}_\Gamma^{(k)}$$

for a given initial value $\bar{u}_\Gamma^{(0)} := [0,\, R_\Gamma^T]\, u^{(0)}$, $u^{(0)} \in W$. We obtain the update $\delta\bar{u}_\Gamma^{(k)}$ by solving the linear system

$$R_\Gamma^T DS(R_\Gamma\bar{u}_\Gamma^{(k)})R_\Gamma\,\delta\bar{u}_\Gamma^{(k)} = R_\Gamma^T S(R_\Gamma\bar{u}_\Gamma^{(k)}) \tag{2.95}$$

or, using the notation from (2.88),

$$S_g(h(R_\Gamma\bar{u}_\Gamma^{(k)}),R_\Gamma\bar{u}_\Gamma^{(k)})\,\delta\bar{u}_\Gamma^{(k)} = R_\Gamma^T S(R_\Gamma\bar{u}_\Gamma^{(k)}). \tag{2.96}$$

To obtain the right hand side of (2.96), we have to solve local nonlinear systems. From (2.93) we have

$$R_\Gamma^T S(R_\Gamma \bar{u}_\Gamma^{(k)}) = R_\Gamma^T (K_\Gamma(h(R_\Gamma \bar{u}_\Gamma^{(k)}), R_\Gamma \bar{u}_\Gamma^{(k)}) - f_\Gamma).$$

We thus need to compute the nonlinear extension $u_I^* := h(R_\Gamma \bar{u}_\Gamma^{(k)})$ of the values on the interface by solving

$$K_I(u_I^*, R_\Gamma \bar{u}_\Gamma^{(k)}) - f_I = 0. \tag{2.97}$$

We obtain u_I^* by carrying out the Newton iteration $u_I^{(j+1)} = u_I^{(j)} - \alpha^{(j)} \delta u_I^{(j)}$ with

$$DK(u_I^{(j)}, R_\Gamma \bar{u}_\Gamma^{(k)})_{II}\, \delta u_I^{(j)} = K_I(u_I^{(j)}, R_\Gamma \bar{u}_\Gamma^{(k)}) - f_I. \tag{2.98}$$

We remark that the problems in (2.98) are uncoupled and therefore completely local. In the last step we apply the linear BDDC preconditioner $M_{BDDC}^{-1}(u_I^*, R_\Gamma \bar{u}_\Gamma^{(k)})$ to equation (2.96), cf. also (2.89), and solve with some Krylov iteration. We also need $h(R_\Gamma \bar{u}_\Gamma^{(k)}) = u_I^*$ to obtain $S_g(h(R_\Gamma \bar{u}_\Gamma^{(k)}), R_\Gamma \bar{u}_\Gamma^{(k)})$ in (2.96).

With the equivalence of (2.96) and $F_{NL1}(\tilde{u}) = F_{NL2}(\tilde{u}) = F_{NK}(\tilde{u})$ for nonlinear FETI-DP, and since we use the standard linear preconditioner, we also have the equality of the eigenvalues of the preconditioned system matrices of the nonlinear FETI-DP and nonlinear BDDC methods except of the eigenvalues zero and one. Of course this is only true if the tangential matrix is built in the same point $\tilde{u}$ for all methods. But as a result of the different right hand sides the nonlinear methods are indeed very different from each other and can show a drastically different convergence behavior. Also note that in our nonlinear FETI-DP methods the coarse problem can improve the Newton convergence whereas in Nonlinear-BDDC this is not the case, since the coarse space is only included in the linear preconditioner applied to the linearized system.

2.6.4 Comparison of the Nonlinear FETI-DP Methods and Nonlinear-BDDC

Since the coarse problem is included in the operator in the nonlinear FETI-DP methods, the coarse space has a direct influence on the Newton iteration, in addition to the Krylov subspace iteration. This is not the case for the nonlinear BDDC method. Since the primal assembly is only part of the linear preconditioner in Nonlinear-BDDC, the choice of the coarse space has no effect on the Newton iterations but only on the convergence of the Krylov space method.

In the nonlinear formulations that we choose for FETI-DP and BDDC, both algorithms therefore have a quite different convergence behavior.

Note that the local nonlinear problems in the Nonlinear-BDDC method are completely decoupled whereas in the nonlinear FETI-DP methods the local nonlinear problems are coupled in the primal variables. This increases the communication cost but improves the convergence. Note that an additional coarse space can be applied to the linearized problems, e.g., using deflation; see, e.g., [88].

The coarse space of the nonlinear FETI-DP methods is nonlinear whereas in the Nonlinear-FETI-1 approach by Pebrel, Rey, and Gosselet [117] the coarse space is linear. From a theoretical viewpoint, Nonlinear-BDDC relies on similar assumptions as the nonlinear FETI-DP methods, see Section 2.5.1.

We may also construct nonlinear FETI-DP and BDDC methods by applying a nonlinear preconditioner to the dual or primal nonlinear Schur complement systems. In this case two nonlinear systems would have to be solved on the right hand side of each global Newton iteration, but this is not topic of this thesis.

2.7 A Nonlinear Model Problem and Numerical Results

In this section, we present numerical results for different model problems based on the p-Laplace operator with homogeneous Dirichlet boundary conditions. Results are shown for the two nonlinear FETI-DP methods, for Newton-Krylov-FETI-DP, and for the nonlinear BDDC algorithm. All algorithms are implemented in Matlab and use the same building blocks; see Section 2.4.4 for a description of the building blocks and for pseudocodes of the different algorithms. The results can thus be seen as numerical scalability studies of the different approaches. We will also present a brief outlook on the improved parallel performance and efficiency of the new nonlinear methods compared to the standard Newton-Krylov-FETI-DP approach in Section 2.7.4.

2.7.1 The p-Laplace Equation

Let us define the p-Laplace operator for $p \geq 2$ by

$$\Delta_p u := \operatorname{div}(|\nabla u|^{p-2}\nabla u).$$

A solution of the partial differential equation

$$\begin{aligned} -\Delta_p u &= b && \text{in } \Omega \\ u &= 0 && \text{on } \partial\Omega, \end{aligned} \tag{2.99}$$

minimizes the energy

$$J(u) := \int_{\Omega} \frac{1}{p} |\nabla u|^p - bu \, dx.$$

Using the decomposition of Ω into nonoverlapping subdomains Ω_i, $i = 1,...,N$, we define

$$J_i(u_i) := \int_{\Omega_i} \frac{1}{p} |\nabla u_i|^p - bu_i \, dx \, , i = 1,...,N.$$

This decomposition satisfies Assumption 1; see Section 2.3. We have to compute the derivative of $J_i(u_i)$, see Section 2.3, and obtain

$$J_i'(u_i)(v_i) = \int_{\Omega_i} |\nabla u_i|^{p-2} \nabla u_i^T \nabla v_i - bv_i \, dx. \tag{2.100}$$

Therefore, given a finite element basis $\{\varphi_1, \ldots, \varphi_{N_i}\}$ on a subdomain Ω_i, we have

$$K_i(u_i) := \left(\int_{\Omega_i} |\nabla u_i|^{p-2} \, \nabla u_i^T \nabla \varphi_1 dx \; , \; ... \; , \; \int_{\Omega_i} |\nabla u_i|^{p-2} \nabla u_i^T \nabla \varphi_{N_i} dx \right)^T$$

and the right hand sides

$$f_i := \left(\int_{\Omega_i} b\varphi_1 dx \; , \; ... \; , \; \int_{\Omega_i} b\varphi_{N_i} dx \right)^T.$$

For the tangential matrices $DK^{(i)}(u_i)$, we obtain

$$(DK^{(i)}(u_i))_{j,k} := \int_{\Omega_i} |\nabla u_i|^{p-2} \nabla \varphi_j^T \nabla \varphi_k dx + (p-2) \int_{\Omega_i} |\nabla u_i|^{p-4} \, (\nabla u_i^T \nabla \varphi_j) \, (\nabla u_i^T \nabla \varphi_k) dx$$

directly by computing the derivative of $J_i'(u_i)(v_i)$ in the direction w_i and by using the basis representations of v_i and w_i.

2.7.2 Numerical Results for Nonlinear FETI-DP

We test our algorithms for four different nonlinear model problems based on variants of the p-Laplace equation. First, we consider the standard p-Laplace equation (2.99). The second model problem has nonlinearities which are local with respect to the subdomains. In the third model problem, we have local nonlinearities which are not confined to the subdomains. The fourth model problem is nonlinear on the complete domain but has large discontinuities in the coefficients.

In all experiments in this section, we consider the unit square $\Omega := [0,1] \times [0,1]$ and decompose it into square subdomains $\Omega_i, i = 1,...,N$. We have chosen piecewise linear finite elements. In all tests all vertices are primal and, additionally, all edges are primal, i.e., edge averages are used as primal edge constraints in our linear and nonlinear FETI-DP methods. We carry out experiments using both standard arithmetic edge averages and weighted averages. For a given edge $\mathscr{E}$, we consider edge averages $\overline{u}^{\mathscr{E}}$ of the form

$$\overline{u}^{\mathscr{E}} := \frac{\sum_{i=1}^{N_{\mathscr{E}}} \rho_i u_i}{\sum_{i=1}^{N_{\mathscr{E}}} \rho_i}, \tag{2.101}$$

where $N_{\mathscr{E}}$ is the number of nodes x on the edge $\mathscr{E}$, ρ_i is a weight associated with such a node, and $u_i = u(x)$ is the nodal evaluation of the function to be averaged. For $\rho_i = 1, i = 1, \ldots, N_{\mathscr{E}}$, we have standard arithmetic edge averages. Otherwise, the weights ρ_i are defined by using the coefficients of the differential equations; see equation (2.105). We always use the following stopping criteria. Newton iterations, which solve systems of the form $\widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f} = 0$, see (2.44) and (2.50), are stopped when $||\widetilde{K}(\tilde{u}^{(k)}) + B^T\lambda - \tilde{f}||_{L_2} < 1e{-}12$. Furthermore, in all our FETI-DP algorithms, we stop the global Newton iteration when the fully assembled residual satisfies $||R^T K(R_\Pi \tilde{u}^{(k)}) - R^T f||_{L_2} < 1e{-}12$. Here, R^T is the global assembly operator on the interface from (2.16). Our stopping criterion is thus based on the variable u and not on λ for NL-FETI-DP-1 as well as NL-FETI-DP-2. We thus have identical stopping criteria for all three FETI-DP methods. As initial values we always use

$$\lambda^{(0)} = 0$$

and

$$u^{(0)}(x_1, x_2) = x_1\,(1 - x_1)\,x_2\,(1 - x_2)$$

which satisfies our homogeneous Dirichlet boundary condition. We cannot use $u^{(0)} = 0$ since the tangential matrix is then singular. For $u \neq 0$ the assembled or also partially assembled tangential matrix is always positive definite and the Newton direction a decent direction. In our experiments in this section, we always choose the conjugate gradient method (CG) as a Krylov space method. The stopping criterion for the preconditioned Krylov iterations is the relative reduction of the preconditioned residual to $1e{-}10$. Note that we always use recycled Lagrange multipliers as start values, i.e., we use the resulting Lagrange multiplier of the previous Newton iteration as a start value for the Krylov subspace iteration in the current Newton iteration. If an absolute convergence criterion is used for the CG iteration this simple strategy can lead to a substantial reduction of the number of Krylov iterations.

If a relative convergence criterion is used for the CG iteration, the absolute accuracy of the Krylov solution increases during the Newton iteration, since the

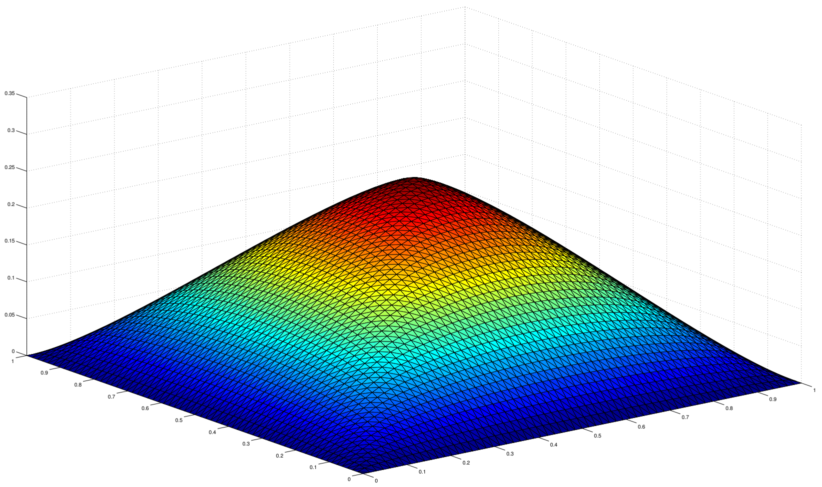

Fig. 2.6: Solution of the p-Laplace equation $-\Delta_4 u = 1$ defined in (2.99). The typical diameter of a finite element is $h = 1/64$ in this case.

residual norm of the initial value decreases by using a recycling of λ. Therefore, a relative reduction of $1e{-}10$ is sufficient for the Krylov iteration. In all our computations the Dirichlet preconditioner was used in the CG iterations.

First Model Problem

Our first model problem is the p-Laplace equation (2.99) for $p = 4$ with a constant right hand side $b = 1$. A visualization of the solution can be found in Fig. 2.6. We present some numerical scalability results for the two nonlinear FETI-DP methods and for Newton-Krylov-FETI-DP in Table 2.1 (right) and Table 2.2 (right). Here, we have $H/h = 16$, i.e. $2(16^2)$ finite elements per subdomain, and we increase the number of subdomains. In the usual linear setting, a domain decomposition method is considered numerically scalable, if, for an increasing number of subdomains, the number of iterations is asymptotically bounded. Generally, numerical scalability of a domain decomposition method cannot be obtained without incorporating a coarse problem to accelerate convergence, by ensuring a global transport of information. Numerical scalability is usually regarded as necessary to obtain weak parallel scalability of a domain decomposition method.

In our nonlinear FETI-DP methods the coarse problem is constructed from local constraints as in case of the linear method and will accelerate the Krylov iteration. But an important property of the nonlinear FETI-DP methods is the possibility to accelerate also the Newton convergence by the choice of the coarse problem.

From the results in Table 2.1 (right) we see that for all methods we obtain Newton convergence using our initial value $u^{(0)}$. Moreover, the performance of the two nonlinear FETI-DP methods is similar. Note that for our nonlinear FETI-DP methods, we seem to have numerical scalability in terms of the global number of Krylov iterations. This is a strong statement since this requires the scalability of the Krylov iteration as well as of the Newton iteration.

The most important result is that using the nonlinear FETI-DP methods the number of the global Krylov iterations is reduced by a factor of between 4 and 9 compared to the Newton-Krylov-FETI-DP approach. In a parallel environment the global communication in the Krylov method can thus be reduced significantly. At the same time the number of factorizations is always higher in the nonlinear methods, i.e., by a factor of up to 1.5. This represents an increase of local computational work on the subdomains and is expected.

In general, Newton's method will not converge without some kind of load stepping or globalization strategy. We therefore also report on results using the Wolfe step optimization in all methods; see Table 2.2 (right). Indeed, the Newton-Krylov-FETI-DP approach profits most from the optimized step length. Nevertheless, using the nonlinear methods the number of global Krylov space iterations is still reduced by a factor of between 2 and 4.

Second Model Problem

By design, we expect our new methods to perform best for problems with strong localized nonlinearities. In order to analyze problems with such local nonlinearities we consider the p-Laplace equation with different values of p in different parts of the computational domain as a second model problem. Note that the p-Laplace operator for $p = 2$ is the standard, linear Laplacian. More precisely, we consider a matrix material, where in each subdomain an inclusion is embedded; see Fig. 2.7 (left). Each inclusion is surrounded by a hull of width η of the matrix material. We consider the p-Laplace operator with $p = 2$ in the hull and with $p = 4$ in the inclusions. This configuration can be seen as a nonlinear analog to the problems in [58].

Let us denote the hull inside each subdomain Ω_i by

$$\Omega_{i,\eta} := \{x \in \Omega_i \,:\, \mathrm{dist}(x, \partial\Omega_i) < \eta\}$$

and the inclusion by

$$\Omega_{i,I} := \Omega_i \setminus \Omega_{i,\eta}.$$

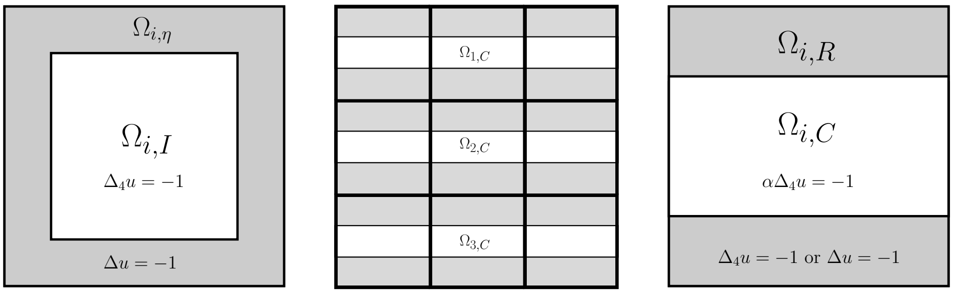

Fig. 2.7: Left: Subdomain Ω_i with an inclusion $\Omega_{i,I}$ surrounded by a hull $\Omega_{i,\eta}$ with width $\eta = \frac{H}{8}$. Middle: Example for a decomposition in $N = 9$ subdomains, intersected by 3 channels $\Omega_{i,C}, i = 1,2,3$. Right: Subdomain Ω_i with channel $\Omega_{i,C}$ of width $\frac{H}{2}$; published in [76].

Furthermore we define the union of all inclusions by

$$\Omega_I := \bigcup_{i=1}^{N} \Omega_{i,I}$$

and the union of all hulls by

$$\Omega_\eta := \bigcup_{i=1}^{N} \Omega_{i,\eta}.$$

We then consider

$$\begin{array}{rcll} -\alpha\Delta_p u - \beta\Delta u & = & 1 & \text{in } \Omega \\ u & = & 0 & \text{on } \partial\Omega, \end{array} \tag{2.102}$$

where $\alpha, \beta : \Omega \to \mathbb{R}$ are coefficient functions given by

$$\alpha(x) = \begin{cases} 1 & \text{if } x \in \Omega_I \\ 0 & \text{elsewhere} \end{cases} \qquad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_I \\ 1 & \text{elsewhere;} \end{cases} \tag{2.103}$$

see Fig. 2.8 for a plot of the solution.

We again carry out numerical experiments to analyze the numerical scalability of the different methods; see Table 2.1 (left) and Table 2.2 (left). Again, in the nonlinear variants significantly fewer Krylov iterations are needed in comparison to NK-FETI-DP, indicating a considerable reduction of the communication in a parallel environment. If the Wolfe step strategy is used in all methods the number of Krylov iterations is reduced by a factor of between 3 and 4 compared to classical NK-FETI-DP, see Table 2.2 (left).

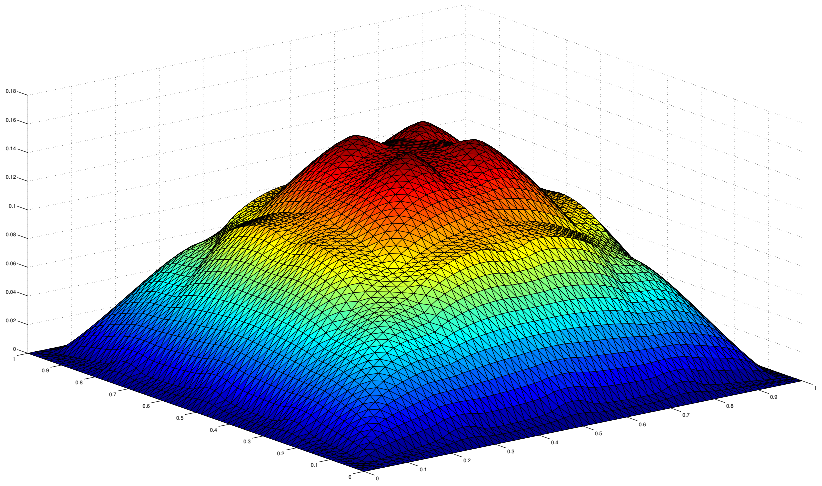

Fig. 2.8: Solution of model equation defined in (2.102) with coefficients defined in (2.103). The domain Ω is decomposed into 16 subdomains and all finite elements have a typical diameter of $h = 1/64$.

For comparison, we also provide the results for the case when no globalization technique is used. In this case the number of Krylov iterations can be reduced by a factor of 4 to 11 when using the nonlinear methods; see Table 2.1 (left). Again, the number of factorizations is higher in the new methods, as expected. We also provide a visual comparison of the convergence behavior of NK-FETI-DP and NL-FETI-DP-2 for this kind of model problem in Fig. 2.2.

Third Model Problem

As a third model problem, we consider localized nonlinearities which are not confined to the subdomains. Instead of inclusions, we now consider channels of width $\frac{H}{2}$ which intersect rows of subdomains from the left boundary to the right boundary of Ω. All channels are parallel and each subdomain is intersected by exactly one channel; see Figure 2.7 (middle and right). We have $p = 2$ in the matrix material outside of the channels and $p = 4$ in the channels. In addition, we scale the p-Laplace operator in the channels with a large coefficient $\tilde{\alpha}$; see (2.104). Let us denote by Ω_C the union of the channels $\Omega_{i,C}$, $i = 1,...,N^{\frac{1}{2}}$, and the rest of the domain by $\Omega_R := \Omega \setminus \Omega_C$; see Figure 2.7 (left and middle). We consider problem (2.102) with coefficient functions $\alpha, \beta : \Omega \to \mathbb{R}$

$$\alpha(x) = \begin{cases} \tilde{\alpha} & \text{if } x \in \Omega_C \\ 0 & \text{elsewhere} \end{cases} \qquad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_C \\ 1 & \text{elsewhere,} \end{cases} \tag{2.104}$$

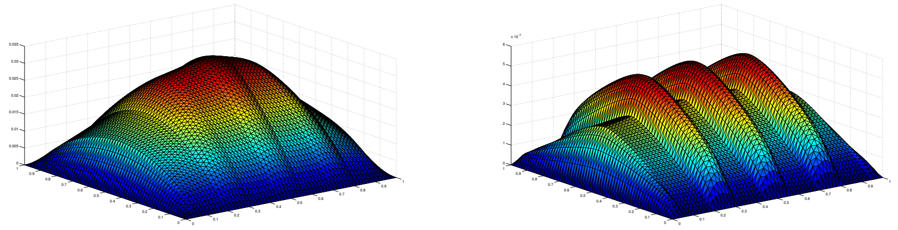

Fig. 2.9: Solution of the equation defined in (2.102) with coefficients defined in (2.104) with with $\tilde{\alpha} = 10^3$ (left) and $\tilde{\alpha} = 10^6$ (right). The domain Ω is decomposed into 16 subdomains and all finite elements have the typical diameter $h = 1/64$.

where $\tilde{\alpha} \in \{10^3, 10^6\}$; see Fig. 2.9 for the solution.

Again, we test for numerical scalability. In a first set of experiments with $H/h = 16$, only standard arithmetic edge averages are used for the primal edges; see Table 2.3. We see the typical behavior, i.e., in the nonlinear methods the number of Krylov iterations is reduced but the number of factorizations, i.e., the local work, is increased. We also see that increasing the coefficient $\tilde{\alpha}$ from 10^3 to 10^6 affects all methods, i.e., the number of Krylov iteration grows. For the first time, we also see significant differences in the maximal condition numbers of the linearized systems. In the NK-FETI-DP approach, for $\tilde{\alpha} = 10^6$, systems with very high condition numbers appear whereas in the nonlinear methods the condition numbers remain small.

Next, we consider a coarse space with weighted edge averages. The scaling is inspired by the weights used in edge averages of linear FETI-DP methods for problems with jumps not aligned to the interface; see [85].

Here, the weights in the weighted edge averages, see (2.101), are defined as

$$\rho_i := \frac{\int_{\mathscr{E}} (\alpha(x) + \beta(x)) \varphi_i(x) dx}{\int_{\mathscr{E}} \varphi_i(x) dx} \tag{2.105}$$

in each degree of freedom x_i on the edge $\mathscr{E}$ with corresponding basis function φ_i. The numerical scalability results are presented in Table 2.4. Again, the nonlinear methods reduce the number of Krylov iterations significantly, i.e., by a factor of between 2 and 7. The local work, as measured by the number of factorizations, is increased by a factor of up to 1.5. Comparing Table 2.3 to Table 2.4, we see that the scaling improves the results of all methods. The results from Table 2.4 also indicate that for the nonlinear methods, i.e., Nonlinear-FETI-DP-1 and Nonlinear-FETI-DP-2, we have robustness of the number of Krylov iterations with respect to the coefficient $\tilde{\alpha}$. This does not seem to be the case for NK-

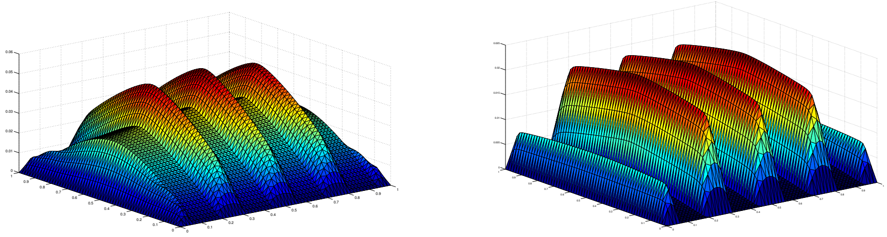

Fig. 2.10: Solution of the equation defined in (2.102) with coefficients defined in (2.106) with $\tilde{\alpha} = 10^3$ (left) and $\tilde{\alpha} = 10^6$ (right). The domain Ω is decomposed into 16 subdomains and all finite elements have the typical diameter $h = 1/64$.

FETI-DP, here the total number of Krylov iterations increases. Note that at the same time the maximal condition number does not increase. The higher number of Krylov iterations is thus a result of slower Newton convergence in NK-FETI-DP.

Fourth Model Problem

Finally, in our fourth model problem, we present results for a nonlinear problem where some methods do not converge without a linesearch.

We have $p = 4$ in the whole domain but we have high contrasts in the coefficient α between the channels and the matrix material. We have

$$\alpha(x) = \begin{cases} \tilde{\alpha} & \text{if } x \in \Omega_C \\ 1 & \text{elsewhere} \end{cases} \qquad \beta(x) = 0 \text{ in } \Omega, \tag{2.106}$$

with $\tilde{\alpha} \in \{10^3, 10^6\}$; see Fig. 2.10 for a plot of the solution. We present numerical scalability results in Tables 2.5 and 2.6. Once more we see a reduction of the number of Krylov iterations in case of a well chosen coarse space, see Table 2.6, where weighted edge averages are used; cf. (2.105). It can also be seen that NL-FETI-DP-2 needs a good coarse space to converge, or, alternatively, an optimization of the step length as introduced in Section 2.4.4. The latter increases the computational cost by some additional factorizations but increases the convergence radius and reduces the number of Krylov iterations; see Table 2.5.

Summary of the Numerical Results

From all our experiments with nonlinear FETI-DP, we can conclude that the nonlinear algorithms always reduce the number of Krylov iterations in comparison to NK-FETI-DP. The reduction can be substantial, i.e., up to 80% if we use Wolfe step length optimization and up to 90% if no globalization technique is used. This will be an important advantage in highly nonlinear parallel simulations since a reduction of the number of Krylov iterations will also reduce the amount of communication between processors and nodes. Notably, the new methods show superior performance, with respect to the total number of Krylov iterations, not only for problems with localized nonlinearities but also for the standard p-Laplace problem.

For some problems, see the third and fourth model problem above, a sufficiently good coarse space has to be chosen. Then, we can also save Newton iterations and factorizations. It should be pointed out that in some cases, without a carefully chosen coarse space, the NL-FETI-DP-2 method does not converge without using an appropriate line search method. This is not a drawback since, in general, suitable globalization techniques are necessary for Newton-based iterative algorithms when applied to nonlinear problems. It also merely reflects the additional flexibility to influence the Newton convergence by the nonlinear FETI-DP coarse space. The choice of a good coarse space is thus even more vital than in linear problems. Our results suggest that known strategies for the construction of problem specific coarse spaces may carry over from the linear case. Finally, let us point out that in some cases the nonlinear algorithms clearly have smaller condition numbers than NK-FETI-DP.

		p-Laplace inclusions				p-Laplace			
N	Solver	#Krylov It.	#Factor.	max. cond.	min. cond.	#Krylov It.	#Factor.	max. cond.	min. cond.
4	NK-FETI-DP	33	14	1.0048	1.0001	72	18	1.1352	1.0608
	NL-FETI-DP-2	5	14	1.2813	1.0000	8	19	1.0644	1.0604
	NL-FETI-DP-1	5	15	1.2805	1.0001	12	20	1.0644	1.0604
16	NK-FETI-DP	105	15	1.4719	1.2914	164	20	1.4605	1.4107
	NL-FETI-DP-2	21	18	1.4240	1.4233	32	29	1.4208	1.4012
	NL-FETI-DP-1	28	18	1.4240	1.4233	40	24	1.4208	1.4108
64	NK-FETI-DP	164	17	1.5680	1.4264	226	22	1.5302	1.4895
	NL-FETI-DP-2	30	20	1.5255	1.5197	52	33	2.1258	1.4878
	NL-FETI-DP-1	40	20	1.5254	1.5197	52	26	2.1258	1.4850
256	NK-FETI-DP	190	19	1.5852	1.5281	268	24	1.6846	1.5394
	NL-FETI-DP-2	31	22	1.5643	1.5412	44	34	2.1523	1.5237
	NL-FETI-DP-1	42	22	1.5654	1.5406	55	28	2.1523	1.5375
1024	NK-FETI-DP	209	21	1.5786	1.4939	293	26	1.9809	1.5642
	NL-FETI-DP-2	31	24	1.5827	1.5409	45	35	2.1669	1.4921
	NL-FETI-DP-1	43	24	1.5852	1.5409	56	30	2.1669	1.5560
4096	NK-FETI-DP	215	23	1.5784	1.4972	330	28	2.5309	1.5657
	NL-FETI-DP-2	19	25	1.5768	1.5451	45	37	2.1743	1.4890
	NL-FETI-DP-1	41	26	1.5938	1.5451	45	31	2.1743	1.5588

Table 2.1: Numerical scalability for the different methods. The p-Laplace problem described in (2.99) and the p-Laplace inclusions' problem is described in (2.102). For "p-Laplace inclusions" see also Fig. 2.7 (left). Here we used vertex constraints and arithmetic edge averages as primal constraints. In both problems we have $\frac{H}{h} = 16$; N is the number of subdomains; all tests are done without any step length optimization; *#Krylov It.* gives the sum of all Krylov iterations; *#Factor.* gives the total number of factorizations of $D\widetilde{K}$ required, see also Section 2.4.4; *min./max. cond* gives the minimal and maximal condition number of the FETI-DP systems; published in [76].

		p-Laplace inclusions				p-Laplace			
N	Solver	#Krylov It.	#Factor.	max. cond.	min. cond.	#Krylov It.	#Factor.	max. cond.	min. cond.
4	NK-FETI-DP	16	8	1.0000	1.0000	32	8	1.1243	1.0255
	NL-FETI-DP-2	5	8	1.2819	1.0000	8	12	1.0644	1.0604
	NL-FETI-DP-1	5	9	1.2809	1.0000	12	10	1.0644	1.0604
16	NK-FETI-DP	77	11	1.4719	1.4166	96	12	1.4605	1.3919
	NL-FETI-DP-2	21	14	1.4240	1.4235	32	23	1.4208	1.4012
	NL-FETI-DP-1	28	14	1.4240	1.4233	40	16	1.4208	1.4050
64	NK-FETI-DP	117	12	1.5680	1.5079	122	12	1.5196	1.4895
	NL-FETI-DP-2	30	17	1.5256	1.5197	52	27	2.1258	1.4910
	NL-FETI-DP-1	40	16	1.5254	1.5197	52	16	2.1258	1.4896
256	NK-FETI-DP	109	11	1.5808	1.5162	143	13	1.5521	1.5306
	NL-FETI-DP-2	30	15	1.5520	1.5515	45	24	2.1905	1.5409
	NL-FETI-DP-1	42	15	1.5653	1.5405	56	18	2.1905	1.5470
1024	NK-FETI-DP	118	12	1.5814	1.5428	155	14	1.7908	1.5655
	NL-FETI-DP-2	30	17	1.5428	1.5407	45	25	2.2061	1.5657
	NL-FETI-DP-1	42	16	1.5853	1.5407	45	19	2.2061	1.5633
4096	NK-FETI-DP	128	14	1.5792	1.5464	167	16	2.4030	1.5626
	NL-FETI-DP-2	18	17	1.5467	1.5464	42	26	2.2138	1.5640
	NL-FETI-DP-1	40	17	1.5932	1.5467	44	20	2.2138	1.5671

Table 2.2: Numerical scalability. Same set of experiments as in Table 2.1, but Wolfe step length optimization has been used additionally; published in [76].

		$\tilde{\alpha} = 10^3$				$\tilde{\alpha} = 10^6$			
N	**Solver**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**
4	NK-FETI-DP	37	8	1.0226	1.0020	85	17	1.0713	1.0022
	NL-FETI-DP-2	10	9	1.0228	1.0184	17	19	1.0719	1.0712
	NL-FETI-DP-1	15	10	1.0303	1.0184	18	19	1.0719	1.0703
16	NK-FETI-DP	65	8	1.9083	1.2425	181	16	902.9105	4.0830
	NL-FETI-DP-2	32	14	1.2433	1.2399	42	22	4.0951	3.8415
	NL-FETI-DP-1	32	11	1.2429	1.2182	52	20	4.1331	3.8415
64	NK-FETI-DP	77	8	3.0447	1.4598	499	19	2065.0183	7.4192
	NL-FETI-DP-2	35	14	1.4603	1.3811	80	24	7.4483	6.5773
	NL-FETI-DP-1	34	11	1.4657	1.3811	144	22	7.6288	6.5773
256	NK-FETI-DP	101	9	4.3795	1.6339	1062	19	3391.8105	9.3041
	NL-FETI-DP-2	40	14	1.6355	1.5248	139	28	9.3917	8.1879
	NL-FETI-DP-1	40	11	1.6466	1.5248	231	22	9.6438	8.1879
1024	NK-FETI-DP	111	9	5.6480	1.7906	1468	18	4664.6562	10.8297
	NL-FETI-DP-2	47	15	1.7906	1.6541	151	37	10.9412	9.5575
	NL-FETI-DP-1	70	13	1.8105	1.6541	272	22	11.2717	9.5575

Table 2.3: Numerical scalability for the different algorithms for problems with a high contrast in coefficients. Results for (2.102) with α and β defined as in (2.104). We have $p = 4$ and a scaling $\tilde{\alpha} \in \{10^3, 10^6\}$ in the channels. The matrix material is linear since here we have $p = 2$. We have $\frac{H}{h} = 16$. The primal set consists of vertex constraints and arithmetic averages on all edges; N is the number of subdomains; *#Krylov It.* gives the sum of all Krylov iterations; *#Factor.* gives the total number of factorizations of $D\widetilde{K}$ required, see also Section 2.4.4; *min./max. cond* gives the minimal and maximal condition number of the FETI-DP systems; published in [76].

		$\tilde{\alpha} = 10^3$				$\tilde{\alpha} = 10^6$			
N	**Solver**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**
4	NK-FETI-DP	46	8	1.1915	1.1580	88	17	1.1363	1.1004
	NL-FETI-DP-2	12	9	1.1915	1.1916	18	19	1.1116	1.1108
	NL-FETI-DP-1	18	10	1.1916	1.1904	18	19	1.1201	1.1116
16	NK-FETI-DP	63	8	1.2588	1.1693	115	16	1.1502	1.1541
	NL-FETI-DP-2	24	12	1.2588	1.2114	21	18	1.1539	1.1541
	NL-FETI-DP-1	32	11	1.2582	1.2114	21	18	1.1551	1.1523
64	NK-FETI-DP	70	8	1.3414	1.2161	141	19	1.1609	1.1507
	NL-FETI-DP-2	27	12	1.3414	1.3397	21	21	1.1591	1.1578
	NL-FETI-DP-1	35	11	1.3526	1.2726	21	21	1.1601	1.1578
256	NK-FETI-DP	81	9	1.4586	1.3532	152	19	1.1716	1.1502
	NL-FETI-DP-2	36	13	1.3977	1.3506	24	21	1.1716	1.1704
	NL-FETI-DP-1	36	11	1.4426	1.3503	22	21	1.1733	1.1704
1024	NK-FETI-DP	75	8	1.7873	1.3626	144	18	1.1907	1.1505
	NL-FETI-DP-2	42	14	2.4643	1.5127	24	21	1.2794	1.1907
	NL-FETI-DP-1	40	11	1.5115	1.5000	22	21	1.1910	1.1819
4096	NK-FETI-DP	102	10	2.0980	1.4107	142	18	1.2317	1.1508
	NL-FETI-DP-2	52	19	3.6975	1.6316	25	20	1.3335	1.2036
	NL-FETI-DP-1	43	14	1.6462	1.5998	23	20	1.2227	1.1760

Table 2.4: Numerical scalability. Same set of experiments as in Table 2.3 but we used weighted edge averages instead of arithmetic ones; published in [76].

		without LS				with LS			
N	Solver	#Krylov It.	#Factor.	max. cond.	min. cond.	#Krylov It.	#Factor.	max. cond.	min. cond.
4	NK-FETI-DP	117	18	1.1647	1.0230	82	13	1.1646	1.0230
	NL-FETI-DP-2	18	20	1.0902	1.0691	18	15	1.0902	1.0691
	NL-FETI-DP-1	18	20	1.0903	1.0691	18	15	1.0903	1.0691
16	NK-FETI-DP	361	20	30277.9739	19.3219	237	15	30277.9739	12.4992
	NL-FETI-DP-2	no	conv.	-	-	95	56	21.8174	4.3789
	NL-FETI-DP-1	141	29	36.2894	4.3789	141	28	36.2892	4.3789
64	NK-FETI-DP	1317	22	50186.1358	58.1841	961	19	50186.1358	34.6106
	NL-FETI-DP-2	no	conv.	-	-	220	71	68.4028	7.1685
	NL-FETI-DP-1	517	35	250.9781	7.1685	513	33	250.9749	7.1685

Table 2.5: Numerical scalability results for (2.102) with α and β defined as in (2.106). We have $p = 4$ in the whole domain and in the channels a coefficient $\tilde{\alpha} = 10^6$; "without LS" means that no step length optimization is used and "with LS" means that we use Wolfe conditions in all Newton type iterations, except the interface problem of Nonlinear-FETI-DP-2, where we use algorithm from Section 3.4.2. We have $\frac{H}{h} = 16$. The coarse space consists of vertex constraints and arithmetic edge averages; N is the number of subdomains; *#Krylov It.* gives the sum of all Krylov iterations; *#Factor.* gives the total number of factorizations of $D\widetilde{K}$ required, see also Section 2.4.4; *min./max. cond* gives the minimal and maximal condition number of the FETI-DP systems; published in [76].

		without LS				with LS			
N	Solver	#Krylov It.	#Factor.	max. cond.	min. cond.	#Krylov It.	#Factor.	max. cond.	min. cond.
4	NK-FETI-DP	121	18	1.1644	1.0259	92	14	1.1708	1.0259
	NL-FETI-DP-2	18	20	1.0905	1.0689	18	17	1.0905	1.0689
	NL-FETI-DP-1	18	20	1.0902	1.0689	18	17	1.0902	1.0689
16	NK-FETI-DP	270	20	67.6310	1.0923	181	14	142.3859	1.0923
	NL-FETI-DP-2	18	22	1.0926	1.0650	18	16	1.0926	1.0650
	NL-FETI-DP-1	18	22	1.0935	1.0650	18	16	1.0935	1.0650
64	NK-FETI-DP	523	22	492.5349	1.0937	354	14	467.6129	1.0937
	NL-FETI-DP-2	18	24	1.0939	1.0670	18	19	1.0939	1.0670
	NL-FETI-DP-1	20	24	1.0948	1.0670	20	19	1.0948	1.0670

Table 2.6: Numerical scalability. Same set of experiments as in Table 2.5, but using weighted edge averages instead of arithmetic ones; 'without LS" means that no step length optimization is used and "with LS" means that we use Wolfe conditions in all Newton type iterations, except in the interface problem of Nonlinear-FETI-DP-2, where we do not use any globalization strategy this time; published in [76].

2.7.3 Numerical Results for Nonlinear-BDDC

Here, we present numerical results for Nonlinear-BDDC applied to some of the model problems from the previous subsection. We restrict ourselves to two of our model problems.

First, in Table 2.7, we present some results for the first model problem; see Section 2.7.2. The setting is the same as in the experiments for Table 2.1. It can be seen that in Nonlinear-BDDC a higher number of Krylov iterations and factorizations is needed compared to Nonlinear-FETI-DP-2. Note that the solution of the local nonlinear problems (2.97) that appear on the right hand side of (2.96) are completely local to the subdomains and thus have a lower computational cost than the problems (2.44) and (2.50) that appear in nonlinear FETI-DP methods. Let us also note that the nonlinear BDDC and the nonlinear FETI-DP algorithms have a difference in the coarse space. The coarse space of the nonlinear FETI-DP algorithms is nonlinear whereas that of the nonlinear BDDC methods is linear. This might explain the difference in the number of Krylov iterations of the two methods.

We also present numerical results for Nonlinear-BDDC applied to the third model problem; see Section 2.7.2. The setting is the same as in the experiments for Table 2.4. Again, a higher number of Krylov iterations and factorizations can be seen for Nonlinear-BDDC in comparison to Nonlinear-FETI-DP-2.

		p-Laplace inclusions				**p-Laplace**			
N	**Solver**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**
4	NL-FETI-DP-2	5	14	1.2813	1.0000	8	19	1.0644	1.0604
	NL-BDDC	16	23	1.2939	1-2891	31	31	1.8322	1.0585
16	NL-FETI-DP-2	21	18	1.4240	1.4233	32	29	1.4208	1.4012
	NL-BDDC	40	28	1.4582	1.4238	84	40	2.4491	1.4176
64	NL-FETI-DP-2	30	20	1.5255	1.5197	52	33	2.1258	1.4878
	NL-BDDC	44	30	1.5479	1.5110	109	45	3.1721	1.4960
256	NL-FETI-DP-2	31	22	1.5643	1.5412	44	34	2.1523	1.5237
	NL-BDDC	43	32	1.5687	1.5145	136	51	9.5875	1.5358
1024	NL-FETI-DP-2	31	24	1.5827	1.5409	45	35	2.1669	1.4921
	NL-BDDC	41	34	1.5750	1.5311	227	54	278.1628	1.5420
4096	NL-FETI-DP-2	19	25	1.5768	1.5451	45	37	2.1743	1.4890
	NL-BDDC	36	36	1.5415	1.5160	229	56	306.0650	1.5706

Table 2.7: Comparison of nonlinear FETI-DP and nonlinear BDDC methods. Same problem setting as in Table 2.1; published in [76].

2.7.4 First Parallel Results

In order to verify if we can also benefit from our new nonlinear FETI-DP approaches in parallel computations, we will provide some first parallel weak scalability results in this section. All algorithms are implemented in PETSc [4–6]

		$\tilde{\alpha} = 10^3$				$\tilde{\alpha} = 10^6$			
N	**Solver**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**	**#Krylov It.**	**#Factor.**	**max. cond.**	**min. cond.**
4	NL-FETI-DP-2	12	9	1.1915	1.1916	18	19	1.1116	1.1108
	NL-BDDC	40	22	1.1896	1.0828	90	51	1.1206	1.0843
16	NL-FETI-DP-2	24	12	1.2588	1.2114	21	18	1.1539	1.1541
	NL-BDDC	78	23	3.9861	1.2292	217	57	2.8897	1.1596
64	NL-FETI-DP-2	27	12	1.3414	1.3397	21	21	1.1591	1.1578
	NL-BDDC	75	24	2.3734	1.2761	227	68	10.1874	1.1676
256	NL-FETI-DP-2	36	13	1.3977	1.3506	24	21	1.1716	1.1704
	NL-BDDC	75	25	2.1582	1.3116	223	69	9.2638	1.1850
1024	NL-FETI-DP-2	42	14	2.4643	1.5127	24	21	1.2794	1.1907
	NL-BDDC	77	26	1.9539	1.4420	211	71	8.9207	1.2115
4096	NL-FETI-DP-2	52	19	3.6975	1.6316	25	20	1.3335	1.2036
	NL-BDDC	90	35	2.1861	1.4748	207	69	7.9532	1.2531

Table 2.8: Comparison of nonlinear FETI-DP and nonlinear BDDC methods. Same problem setting as in Table 2.4; published in [76].

using the same building blocks and thus we obtain a fair comparison. We use the same sequential sparse direct solver to factorize matrices K_{BB} and $\widetilde{S}_{\Pi\Pi}$ after broadcasting a serial copy of $\widetilde{S}_{\Pi\Pi}$ to all ranks. Of course, this is a well known scalability limit in exact FETI-DP methods, since the coarse space grows with the number of subdomains and computational cores. A strategy to overcome this scalability limit based on the parallel and inexact solution of the FETI-DP coarse problem is therefore presented in the next chapter. We will also provide a more detailed description of our parallel implementations in Section 3.3. With this in mind, the results presented here can be seen as a motivation or preview for the upcoming chapters.

We again consider inclusions of p-Laplace in a matrix material of linear Laplace. Thus, the setup is largely identical to the *p-Laplace inclusions* setup introduced in Table 2.1, but we consider a higher problem resolution with $H/h = 160$ and iterate to an accuracy of $1e-8$ instead of $1e-12$. We present weak scalability starting with 64 cores of the JUQUEEN BlueGene/Q machine at Forschungszentrum Jülich and scale up to 16384 cores in Table 2.9. The parallel results show a similar numerical behavior as our serial although the subdomain size is now approximately hundred times larger, with $H/h = 160$. Again the number of Krylov iterations is reduced in the nonlinear methods and a slightly higher number of factorizations of K_{BB} and $\widetilde{S}_{\Pi\Pi}$ is necessary. As expected in our discussion of the sequential results this directly affects the time to solution and the parallel performance. Choosing the shortest runtime on 64 cores as a baseline ($61.93s$ of Nonlinear-FETI-DP-2) we finally obtain a parallel efficiency of 40% using Newton-Krylov-FETI-DP, 67% using Nonlinear-FETI-DP-2, and an optimum of 71% using Nonlinear-FETI-DP-1 on 16384 cores. On the one hand, this shows the parallel superiority of the nonlinear methods we hoped for, but, on the other hand we see a capital loss of parallel efficiency in all three methods stepping from 4096 cores to 16384 cores. This is caused by an increased factorization time of $\widetilde{S}_{\Pi\Pi}$ of approximately 1 second, which has to be performed *#Factor.* times. This is an expected effect and directly motivates the discussion of inexact and nonlinear FETI-DP methods in the upcoming chapter. Of course, we could also obtain better parallel scalability in exact FETI-DP methods using parallel sparse direct solvers. Also larger subdomain sizes decrease the impact of the coarse direct solve since the ratio between local and global work is increased.

JUQUEEN BlueGene/Q Supercomputer					
N	**Solver**	**#Krylov It.**	**#Factor.**	**Total Time to Solution**	**Parallel Eff.**
64	NK-FETI-DP	365	15	90.34s	69%
	NL-FETI-DP-2	103	19	61.93s	100%
	NL-FETI-DP-1	168	19	71.83s	86%
256	NK-FETI-DP	471	16	102.15s	61%
	NL-FETI-DP-2	125	19	61.56s	101%
	NL-FETI-DP-1	144	18	64.43s	96%
1 024	NK-FETI-DP	520	17	111.11s	56%
	NL-FETI-DP-2	161	19	66.56s	93%
	NL-FETI-DP-1	149	18	65.67s	94%
4 096	NK-FETI-DP	568	18	124.98s	50%
	NL-FETI-DP-2	130	19	67.81s	91%
	NL-FETI-DP-1	123	18	66.16s	94%
16 384	NK-FETI-DP	579	18	157.77s	40%
	NL-FETI-DP-2	132	19	91.92s	67%
	NL-FETI-DP-1	123	18	87.01s	71%

Table 2.9: Setting as in the p-Laplace inclusion experiments from Table 2.1, but here we have $H/h = 160$ and we only consider vertex constraints.

3 Inexact Reduced Nonlinear FETI-DP

In the theory of linear FETI-DP, it is a well known fact that the exact solution of the FETI-DP coarse problem is a limiting factor for the parallel scalability. A possible strategy to increase the parallel efficiency is the usage of parallel sparse direct solvers but nonetheless, the parallel efficiency of the direct solver remains to be a bottleneck on large scales. In [84,87], Klawonn and Rheinbach suggested to move the solution of the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$ to the preconditioner, which facilitates the usage of an inexact solution, e.g., provided by an additional algebraic multigrid (AMG) method. This approach leads to a fast and stable convergence and enables scalability to a larger number of cores.

In this chapter, we combine the highly parallel efficiency of the linear inexact reduced FETI-DP framework [84, 87] with our Nonlinear-FETI-DP-1 method. This new approach was first proposed in a DD22 plenary talk [77] and is denoted irNonlinear-FETI-DP-1 (inexact reduced Nonlinear-FETI-DP-1) method.

We first provide a detailed description of the inexact reduced Nonlinear-FETI-DP-1 method and our parallel implementation. Then we present weak and strong scalability results for different model problems, as, e.g., problems including the p-Laplace operator as well as nonlinear hyperelasticity problems. The computations have been performed on different architectures and the results are partially taken from the submitted paper [78]. Additionally, in Section 3.5, we will introduce another inexact and nonlinear FETI-DP method, which is based on a nonlinear elimination of interior- and interface variables, denoted by inexact reduced Nonlinear-FETI-DP-2.

3.1 Inexact Reduced Nonlinear-FETI-DP-1

As the exact nonlinear FETI-DP methods described in Chapter 2, inexact reduced Nonlinear-FETI-DP-1 is based on the nonlinear FETI-DP master system (2.34). Recall that Newton linearization of (2.34) with respect to $(\tilde{u},\lambda)$ results in the linear system

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u} \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f} \\ B\tilde{u} \end{bmatrix}. \tag{3.1}$$

With the standard FETI-DP notation introduced in Chapter 2, we partition $\delta\tilde{u}$ into primal variables $\delta\tilde{u}_\Pi$ and dual variables δu_B, i.e.,

$$\delta\tilde{u} = \begin{bmatrix} \delta u_B \\ \delta\tilde{u}_\Pi \end{bmatrix}.$$

Since by construction, we have continuity in the primal variables $\delta\tilde{u}_\Pi$, the jump operator can again be written as $B = [B_B \; 0]$, where the first block B_B corresponds to the dual plus interior variables δu_B and the second zero block to the (subassembled) primal variables $\delta\tilde{u}_\Pi$. We can write (3.1) as block system

$$\begin{bmatrix} (D\widetilde{K}(\tilde{u}))_{BB} & (D\widetilde{K}(\tilde{u}))_{\Pi B}^T & B_B^T \\ (D\widetilde{K}(\tilde{u}))_{\Pi B} & (D\widetilde{K}(\tilde{u}))_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta u_B \\ \delta\tilde{u}_\Pi \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} (\widetilde{K}(\tilde{u}))_B + B_B^T\lambda - f_B \\ (\widetilde{K}(\tilde{u}))_\Pi - \tilde{f}_\Pi \\ B_B u_B \end{bmatrix}. \quad (3.2)$$

Assuming that $(D\widetilde{K}(\tilde{u}))_{BB}$ is invertible, we perform one step of block Gauss elimination of u_B and obtain a reduced system

$$\mathscr{A}_r x_r = \mathscr{F}_r, \quad (3.3)$$

with

$$\mathscr{A}_r = \begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -(D\widetilde{K}(\tilde{u}))_{\Pi B}(D\widetilde{K}(\tilde{u}))_{BB}^{-1}B_B^T \\ -B_B(D\widetilde{K}(\tilde{u}))_{BB}^{-1}(D\widetilde{K}(\tilde{u}))_{\Pi B}^T & -B_B(D\widetilde{K}(\tilde{u}))_{BB}^{-1}B_B^T \end{bmatrix}, \quad (3.4)$$

$$\mathscr{F}_r = \begin{bmatrix} ((\widetilde{K}(\tilde{u}))_\Pi - \tilde{f}_\Pi) \; - \; (D\widetilde{K}(\tilde{u}))_{\Pi B}(D\widetilde{K}(\tilde{u}))_{BB}^{-1}\,((\widetilde{K}(\tilde{u}))_B + B_B^T\lambda - f_B) \\ B_B u_B \; - \; B_B(D\widetilde{K}(\tilde{u}))_{BB}^{-1}\,((\widetilde{K}(\tilde{u}))_B + B_B^T\lambda - f_B) \end{bmatrix}, \quad (3.5)$$

and

$$x_r = \begin{bmatrix} \delta\tilde{u}_\Pi \\ \delta\lambda \end{bmatrix}; \quad (3.6)$$

cf. the notation in [84] for linear problems. The Schur complement $\widetilde{S}_{\Pi\Pi}$ in (3.4), i.e.,

$$\widetilde{S}_{\Pi\Pi} = (D\widetilde{K}(\tilde{u}))_{\Pi\Pi} - (D\widetilde{K}(\tilde{u}))_{\Pi B}(D\widetilde{K}(\tilde{u}))_{BB}^{-1}(D\widetilde{K}(\tilde{u}))_{\Pi B}^T \quad (3.7)$$

provides the coarse problem of the method. An elimination of $\tilde{u}_\Pi$ yields the Nonlinear-FETI-DP-1 (NL1) system

$$F_{NL1}\delta\lambda = d_{NL1};$$

c.f., equation (2.41). Here, we will choose to solve the block system (3.4) instead, using the left block-triangular preconditioner

$$\begin{aligned} &\mathscr{B}_{r,L}^{-1} := \mathscr{B}_{r,L}^{-1}(\widehat{S}_{\Pi\Pi}^{-1}, D\widetilde{K}(\tilde{u})_{BB}^{-1}, M^{-1}) := \\ &\begin{bmatrix} \widehat{S}_{\Pi\Pi}^{-1} & 0 \\ -M^{-1}B_B(D\widetilde{K}(\tilde{u})_{BB}^{-1}(D\widetilde{K}(\tilde{u}))_{\Pi B}^T\hat{S}_{\Pi\Pi}^{-1} & -M^{-1} \end{bmatrix}. \end{aligned} \tag{3.8}$$

The application of $\widehat{S}_{\Pi\Pi}^{-1}$ will consist of a few cycles of a parallel AMG method. For $D\widetilde{K}(\tilde{u})_{BB}^{-1}$ we will use concurrent sparse direct solvers on the subdomains. Remark that we no longer have the inverse $\widetilde{S}_{\Pi\Pi}^{-1}$ in the system operator, but an approximate solution $\widehat{S}_{\Pi\Pi}^{-1}$ in the preconditioner. This move of the coarse problem solution to the preconditioner enables the usage of fast and inexact solvers, as, e.g., an AMG method.

In (3.8), M^{-1} is a good, parallel preconditioner for the dual Schur complement. We choose M^{-1} as one of the standard FETI-DP preconditioners, e.g., the Dirichlet preconditioner from equation (2.8). Its application is embarrassingly parallel.

We use a Krylov space method suitable for unsymmetric systems as GMRES (see also Section 1.1.3) since the preconditioner (3.8) is unsymmetric. The use of conjugate gradients requires a symmetric reformulation; see, e.g., [84].
Let us remark that inexact reduced FETI-DP will have convergence bounds of the same quality as exact FETI-DP, if $\widehat{S}_{\Pi\Pi}$ is a good preconditioner for $\widetilde{S}_{\Pi\Pi}$; see [84] for the theoretical aspects. In general, algebraic multigrid will be a sufficient preconditioner for the FETI-DP coarse space, if smooth error vectors lie in the range of the AMG interpolation operator; see, e.g., [15]. This is always the case in our Laplace or p-Laplace model problems, since all classical AMG interpolation operators for scalar PDEs exactly interpolate constant vectors. In our experience also rotations have to be controlled to obtain optimal convergence considering elasticity and hyperelasticity problems . Therefore, we briefly describe the global matrix (GM) approach in Section 4, first introduced by Baker, Kolev, and Meier-Yang in [3], and present some parallel results for linear and nonlinear elasticity problems, where we use GM interpolations in the AMG preconditioner for irNonlinear-FETI-DP-1. In the GM approach chosen smooth error vectors are interpolated exactly without losing the sparsity of the interpolation operator. Let us remark that for the model problems considered in Section 3.4 and in [78] a classical AMG method was always a sufficient preconditioner.

3.2 Computing an Initial Value

A good initial value can be crucial for the convergence of Newton-type methods. A suitable initial value $\tilde{u}^{(0)}$ for the Newton method as presented in Section 3.1 has to be continuous in all primal variables $\tilde{u}_{\Pi}^{(0)}$ and should provide a good local approximation of the given problem; cf. Section 2.4.2. Note that the initial value is allowed to be discontinuous in the dual variables $u_B^{(0)}$. Let us recall from Section 2.4.2 that an initial value can be obtained from solving the nonlinear problem

$$\widetilde{K}(\tilde{u}^{(0)}) = \tilde{f} - B^T \lambda^{(0)}. \tag{3.9}$$

For simplicity, we use $\lambda^{(0)} = 0$. Here, (3.9) requires the solution of concurrent nonlinear subdomain problems which are only coupled in the primal unknowns. The communication is thus limited to the primal variables, i.e., a few unknowns for each MPI rank. This step can also be seen as a nonlinear localization step.

Linearization of (3.9) results in

$$\begin{bmatrix} (D\widetilde{K}(\tilde{u}))_{BB} & (D\widetilde{K}(\tilde{u}))_{\Pi B}^T \\ (D\widetilde{K}(\tilde{u}))_{\Pi B} & (D\widetilde{K}(\tilde{u}))_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} \delta u_B \\ \delta \tilde{u}_{\Pi} \end{bmatrix} = \begin{bmatrix} (\widetilde{K}(\tilde{u}))_B + B_B^T \lambda - f_B \\ (D\widetilde{K}(\tilde{u}))_{\Pi} - \tilde{f}_{\Pi} \end{bmatrix}, \tag{3.10}$$

and a block Gauss elimination of u_B yields the symmetric system

$$\widetilde{S}_{\Pi\Pi}\, \delta \tilde{u} = \tilde{d}_{\Pi}, \tag{3.11}$$

where $\widetilde{S}_{\Pi\Pi}$ is defined as in (3.7) and

$$\tilde{d}_{\Pi} = (DK(\tilde{u}))_{\Pi} - \tilde{f}_{\Pi} - (D\widetilde{K}(\tilde{u}))_{\Pi B} (D\widetilde{K}(\tilde{u}))_{BB}^{-1} \left((\widetilde{K}(\tilde{u}))_B + B_B^T \lambda - f_B \right).$$

We solve (3.11) by a Krylov method using the algebraic multigrid preconditioner $\widehat{S}_{\Pi\Pi}^{-1}$; see (3.8). We will also refer to the initialization phase as *Phase 1* of the algorithm. The solution phase corresponding to Section 3.1 is then referred to as *Phase 2*; cf. Fig. 3.1

3.3 Algorithmic Building Blocks

Although our approach differs substantially from the Newton-Krylov-FETI-DP approach, the building blocks of the new algorithm are largely identical. In Fig. 3.1 the inexact reduced nonlinear FETI-DP algorithm presented in Section 3.1 is summarized and, for comparison, in Fig. 3.2 the more traditional Newton-Krylov irFETI-DP approach is outlined. Both approaches include an inexact solution of the coarse problem using an AMG method.

```
Inexact Reduced Nonlinear-FETI-DP-1

init: $\tilde{u}^{(0)} \in \widehat{W}$, $\lambda^{(0)} = 0$
for $k = 0, ..., convergence$  // Phase 1: Compute initial value
    build: $\widetilde{K}(\tilde{u}^{(k)})$ and $D\widetilde{K}(\tilde{u}^{(k)})$
    iterative Krylov solve for $\delta u_\Pi^{(k)}$ using
    AMG preconditioner $\widehat{S}_{\Pi\Pi}^{-1}$:
    $\widetilde{S}_{\Pi\Pi}\,\delta\tilde{u}_\Pi^{(k)} = \tilde{d}_\Pi$   // see eq. (3.11)
    update:   // see eq. (3.10)
    $\delta u_B^{(k)} := (D\widetilde{K}(\tilde{u}^{(k)}))_{BB}^{-1}\{(\widetilde{K}(\tilde{u}^{(k)}))_B + B_B^T\lambda^{(0)} - f_B - (D\widetilde{K}(\tilde{u}^{(k)}))_{\Pi B}^T\,\delta\tilde{u}_\Pi^{(k)}\}$
    $\delta\tilde{u}^{(k)} := [\delta u_B^{(k)T}, \delta\tilde{u}_\Pi^{(k)T}]^T$
    compute: steplength $\alpha^{(k)}$
    $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\,\delta\tilde{u}^{(k)}$
end
$\tilde{u}^{(0)} := \tilde{u}^{(k+1)}$
for $k = 0, ..., convergence$  // Phase 2: Main iteration loop
    build: $\widetilde{K}(\tilde{u}^{(k)})$, $D\widetilde{K}(\tilde{u}^{(k)})$, and $M^{-1}$
    iterative Krylov solve for $x_r = [\delta\tilde{u}_\Pi^{(k)T}, \delta\lambda^{(k)T}]$ using left
    preconditioner $\mathscr{B}_{r,L}^{-1} := \mathscr{B}_{r,L}^{-1}(\widehat{S}_{\Pi\Pi}^{-1}, D\widetilde{K}(\tilde{u})_{BB}^{-1}, M^{-1})$:
        $\mathscr{A}_r x_r = \mathscr{F}_r$     // (see eq. (3.3))
    update:   // see eq. (3.2)
    $\lambda^{(k+1)} := \lambda^{(k)} - \alpha^{(k)}\,\delta\lambda^{(k)}$
    $\delta u_B^{(k)} := (D\widetilde{K}(\tilde{u}^{(k)}))_{BB}^{-1}\{(\widetilde{K}(\tilde{u}^{(k)}))_B + B_B^T(\lambda^{(k+1)}) - f_B - (D\widetilde{K}(\tilde{u}^{(k)}))_{\Pi B}^T\,\delta\tilde{u}_\Pi^{(k)}\}$
    $\delta\tilde{u}^{(k)} = [\delta u_B^{(k)^T}, \delta\tilde{u}_\Pi^{(k)^T}]^T$
    $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\,\delta\tilde{u}^{(k)}$
end
```

Fig. 3.1: Pseudocode of the **inexact reduced Nonlinear-FETI-DP-1** algorithm. The application of $\widehat{S}_{\Pi\Pi}^{-1}$ will consist of cycles of a parallel AMG method, for $D\widetilde{K}(\tilde{u})_{BB}^{-1}$ uses concurrent forward-backward substitutions of a sparse direct solver on the subdomains. Figure already published in [78].

```
Newton-Krylov Inexact Reduced FETI-DP Method

Init: $\tilde{u}^{(0)} \in \widehat{W}$
for $k = 0, ..., convergence$
    build: $\widetilde{K}(\tilde{u}^{(k)})$, $D\widetilde{K}(\tilde{u}^{(k)})$, and $M^{-1}$
    iterative Krylov solve for $x_r = [\delta\tilde{u}_{\Pi}^{(k)T}, \lambda]$ using left
    preconditioner $\mathscr{B}_{r,L}^{-1} := \mathscr{B}_{r,L}^{-1}(\widehat{S}_{\Pi\Pi}^{-1}, D\widetilde{K}(\tilde{u})_{BB}^{-1}, M^{-1})$:
        $\mathscr{A}_r x_r = \mathscr{F}_r$        // (see eq. (3.3))

    update:
    $\delta u_B^{(k)} := (D\widetilde{K}(\tilde{u}^{(k)}))_{BB}^{-1} \{(\widetilde{K}(\tilde{u}^{(k)}))_B - f_B + B_B^T \lambda - (D\widetilde{K}(\tilde{u}^{(k)}))_{\Pi B}^T \delta\tilde{u}_{\Pi}^{(k)}\}$
    // see eq. (3.2)

    $\delta\tilde{u}^{(k)} = [\delta u_B^{(k)^T}, \delta\tilde{u}_{\Pi}^{(k)^T}]^T$
    $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\, \delta\tilde{u}^{(k)}$
end
```

Fig. 3.2: Pseudocode of the **Newton-Krylov inexact reduced FETI-DP algorithm**. The application of $\widehat{S}_{\Pi\Pi}^{-1}$ will consist of cycles of a parallel AMG method, for $D\widetilde{K}(\tilde{u})_{BB}^{-1}$ uses concurrent forward-backward substitutions of a sparse direct solver on the subdomains; published in [78].

Besides the algorithmic description in Fig. 3.2 we will not give any description of Newton-Krylov irFETI-DP, since it only differs from Newton-Krylov-FETI-DP (Section 2.2) by using inexact reduced FETI-DP to solve the linearized system.

In our implementation, for the solution of linearized sparse systems, we build upon sparse direct and iterative solvers. For linear problems on the subdomains we apply a multifrontal sparse direct solver which is known for its robustness [28]. Cycles of an algebraic multigrid preconditioner (BoomerAMG) [42,68] are applied to the (linearized) global coarse problem $\widetilde{S}_{\Pi\Pi}$. This adds further (algebraic) coarse levels to the method. All packages were interfaced through PETSc [4–6]. We make use of the infrastructure provided by PETSc where possible.

3.3.1 Parallel Application of $(D\widetilde{K})^{-1}$ to a Vector

We have the product representation

$$
\begin{aligned}
D\widetilde{K}^{-1} \quad = \quad & \begin{bmatrix} I & -(D\widetilde{K})_{BB}^{-1}(D\widetilde{K})_{\Pi B}^{T} \\ 0 & I \end{bmatrix} \times \\
& \begin{bmatrix} (D\widetilde{K})_{BB}^{-1} & 0 \\ 0 & \widetilde{S}_{\Pi\Pi}^{-1} \end{bmatrix} \times \\
& \begin{bmatrix} I & 0 \\ -(D\widetilde{K})_{\Pi B}(D\widetilde{K})_{BB}^{-1} & I \end{bmatrix}.
\end{aligned} \tag{3.12}
$$

Here, the block operator

$$
(D\widetilde{K})_{BB}^{-1} = \begin{bmatrix} ((DK^{(1)})_{BB})^{-1} & & 0 \\ & \ddots & \\ 0 & & ((DK^{(N)})_{BB})^{-1} \end{bmatrix} \tag{3.13}
$$

is completely parallel and is implemented by embarrassingly parallel forward-backward substitutions of a sparse direct solver.

For standard FETI-DP methods with an exact solution of the coarse problem the sparse direct solver is also used to factor $\widetilde{S}_{\Pi\Pi}$. This can be performed by first broadcasting $\widetilde{S}_{\Pi\Pi}$ to all ranks before factoring it locally. In this case, in the subsequent Krylov iteration, no communication is necessary for the coarse problem. Alternatively, $\widetilde{S}_{\Pi\Pi}$ can be factored in parallel on a subset of ranks or on separate ranks. Of course, the parallel scalability of this approach is eventually limited by the scalability of the sparse direct solver.

In our inexact reduced Nonlinear-FETI-DP-1 approach the application of $D\widetilde{K}^{-1}$ on a vector is not required in the main iteration loop; see Fig. 3.1 (Phase 2). Instead, only the application of $(D\widetilde{K})_{BB}^{-1}$ to a vector and of the AMG preconditioner $\widehat{S}_{\Pi\Pi}^{-1}$ to a vector is needed in each Krylov iteration step. In our experiments, only one or two V-cycles of the AMG preconditioner are used in each Krylov space iteration.

3.3.2 Building the Coarse Operator $\widetilde{S}_{\Pi\Pi}$

As in multigrid methods it is vital to build the coarse operator efficiently. In parallel multigrid methods this is achieved by carrying out a parallel triple matrix product to build the RAP Galerkin operator, where R and P are the restriction and prolongation operator, respectively, and A is the system matrix. It is well known that repartitioning of the coarse problem is also vital for efficiency. Fortunately, in FETI-DP methods the coarse problem is small compared to the fine level, i.e., it can be smaller by a factor of 10^3 to 10^4 compared to to original problem [83, 87]. FETI-DP and related domain decomposition methods can apply such aggressive coarsening since direct solvers are used on the fine level instead of smoothers.

The FETI-DP coarse operator $\widetilde{S}_{\Pi\Pi}$ can be built from local contributions computed in parallel, i.e.,

$$\widetilde{S}_{\Pi\Pi} = \textstyle\sum_{i=1}^{N} R_{\Pi}^{(i)T} S_{\Pi\Pi}^{(i)} R_{\Pi}^{(i)}, \tag{3.14}$$

where the $S_{\Pi\Pi}^{(i)}$ are distributed among the processor cores. The parallel assembly can be partially overlapped with the parallel computation of the local Schur complement contributions $S_{\Pi\Pi}^{(i)} = (DK^{(i)})_{\Pi\Pi} - (DK^{(i)})_{\Pi B}(DK^{(i)-1})_{BB}(DK^{(i)T})_{\Pi B}$ of the processor cores.

In some of our experiments on the Mira and Vulcan supercomputers, the coarse problem is first distributed over all MPI ranks and then redistributed to a subset of cores, i.e., 16K coarse cores are used for the problems running on 65K and 262K total cores. We also provide results for an improved approach on the Vulcan BlueGene/Q, avoiding the redistribution process; see Sections 3.4.4 and 3.4.5 for details.

It is interesting to note that in our experiments the current implementation of the assembly of the coarse problem starts to be a bottleneck for more than 131K cores. This has not been observed previously.

3.3.3 Parallel Matrix-free Inversion of $B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T$

Systems with a left hand side $B(D\widetilde{K}(\tilde{u}^{(k)}))^{-1}B^T$ are solved using a Krylov subspace method such as GMRES. Nearest neighbor communication is necessary when applying the operators B^T and B to a vector. The application of $D\widetilde{K}(\tilde{u}^{(k)}))^{-1}$ to a vector is implemented as described above in Section 3.3.1.

3.4 Numerical Results

3.4.1 Model Problems

We again consider different model problems based on the nonlinear p-Laplace operator Δ_p and, additionally, on nonlinear hyperelasticity. Besides the model problems described in Section 2.7.2 we use some additional setups.

For scaling runs on the Mira BlueGene/Q, we use the model problem

$$-\Delta u - 4\Delta_p u = 1, \tag{3.15}$$

with $p = 4$, where Δ is the usual Laplace operator and Δ_p the nonlinear p-Laplace operator. Recalling our notation from equation (2.102) in Section 2.7.2, we obtain (3.15) with the coefficient functions

$$\alpha(x) = 4 \quad \text{and} \quad \beta(x) = 1. \tag{3.16}$$

For this problem the number of Newton iterations and the time to solution is small compared to our other model problems. The computations are carried out on the unit square $\Omega = (0,1) \times (0,1)$. For the discretization, we use linear finite elements. The energy is given by

$$J(u) := \int_\Omega \frac{1}{2}|\nabla u|^2 + |\nabla u|^4 - u\, dx.$$

Our next model problem is nonlinear hyperelasticity. We consider a Neo-Hooke material with a soft matrix material and stiff circular inclusions; see Fig. 3.3 for the geometry.

The strain energy density function W [70, 139] is given by

$$W(u) = \frac{\mu}{2}\left(\mathrm{tr}(\mathrm{F}^\mathrm{T}\mathrm{F}) - 3\right) - \mu \ln(\det(\mathrm{F})) + \frac{\lambda}{2}\ln^2(\det(\mathrm{F}))$$

with the Lamé constants

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

and the deformation gradient $F(x) := \nabla\varphi(x)$; here, $\varphi(x) = x + u(x)$ denotes the deformation and $u(x)$ the displacement of x. The energy functional of which stationary points are computed, is given by

$$J(u) = \int_\Omega W(u) - V(u)dx - \int_\Gamma G(u)ds,$$

where $V(u)$ and $G(u)$ are functionals related to the volume and traction forces. In our experiments in 2D, we have the following material parameters E and ν, see Fig. 3.3 for the geometry: In the circular inclusions we have $E = 210\,000$ and in the surrounding matrix material, $E = 210$. We have chosen $\nu = 0.3$ in the complete domain Ω. The nonlinear elasticity problem is discretized with piecewise quadratic finite elements. For our three-dimensional problems, we have chosen a similar setup with stiff spherical inclusions in softer material. Again, we choose $E = 210\,000$ in the inclusions, $E = 210$ inside the surrounding material, and $\nu = 0.3$ in the complete domain.

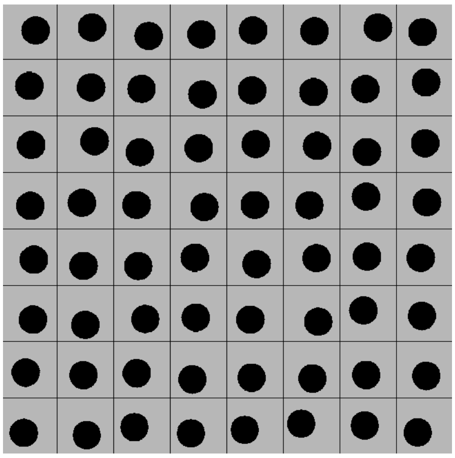

Fig. 3.3: Decomposition of the computational domain Ω into 64 subdomains; each subdomain has an (slightly off-centered) circular inclusion of stiffer material. The problem depicted here is the smallest in Table 3.5.

We always use one iteration of BoomerAMG [2, 42, 68] with symmetric-SOR/Jacobi for the coarse problem. In all our two-dimensional experiments we choose all vertices to be primal. We additionally use edge averages as primal constraints on all edges in three-dimensional problems and perform a local transformation of basis; see [83, Section 4] for details on the implementation of edge constraints.

In all weak scaling experiments in this chapter, the number of subdomains is identical to the number of cores.

3.4.2 Computational Platforms

We perform our computations on four different supercomputers: Mira is a 49 152 node 10-petaflops Blue Gene/Q system at Argonne National Laboratory (USA) with a total number of 786 432 processor cores (Power BQC 16C 1.6GHz; 16 cores and 16 GB memory per node). Mira is ranked No. 5 in the current

TOP500 list (November 2014, www.top500.org). JUQUEEN is also a Blue Gene/Q system at Forschungszentrum Jülich with 458 752 cores and ranked No. 8 in TOP500 list (Power BQC 16C 1.6GHz; 16 cores and 16 GB memory per node). Vulcan is a 24 576 node 5 Petaflop BlueGene/Q production system at Lawrence Livermore National Laboratory (USA) with a total number of 393 216 processor cores (Power BQC 16C 1.6GHz; 16 cores and 16 GB memory per node), and is ranked No. 9 in the current TOP500 list (November 2014). SuperMUC is a 9400 node 3.2-petaflops system at Leibniz Supercomputing Center (Munich, Germany) with a total number of 155 656 processor cores (Sandy Bridge-EP Xeon E5-2680 8C 2.7GHz; 16 cores and 32 GB memory per node). It is ranked No. 14 in the current TOP500 list (November 2014).

On SuperMUC the Intel compiler and on Vulcan and JUQUEEN the IBM XL compiler was used. As we encountered internal compiler errors with the IBM compiler on the BlueGene/Q system in certain templated C++ code we had to resort to the GNU compiler on the Mira supercomputer.

3.4.3 Comparison with Existing Methods

In Table 3.1 a comparison of inexact reduced Nonlinear-FETI-DP-1 with the more traditional Newton-Krylov inexact reduced FETI-DP method [87] is presented. For linear problems both methods are mathematically identical.

We consider the second nonlinear model problem from Section 2.7.2, i.e., the p-Laplace with $p = 4$ in the inclusions embedded inside subdomains and $p = 2$, elsewhere. We have $H/h = 128$ and the inclusions have a distance of $\eta = 2h$ to the subdomain boundary. Here, we use piecewise quadratic finite elements. The runs from 16 up to 65 536 cores have been performed on the Vulcan BlueGene/Q at Lawrence Livermore National Laboratory. By parallel efficiency we refer to the fraction of the actual time to solution compared to the base line, which is, in this case, the time to solution on a single BlueGene/Q node (16 cores). The total time to solution (or total execution time) includes the time for the assembly of the problem as well as its solution. We see a reduction of the number of Krylov iterations and the time used in the Krylov iterations by 70%. As a results, the parallel efficiency of the inexact reduced Nonlinear-FETI-DP-1 method appears to be superior compared to the efficiency of the more traditional Newton Krylov inexact reduced FETI-DP approach. Of course, for these simple nonlinear problems, the time used in the Krylov iteration is small. Nonetheless, the reduction of the total time increases from 28% on 16 cores to 37% on 65 536 cores, i.e., the advantage of the new method increases with the number of cores.

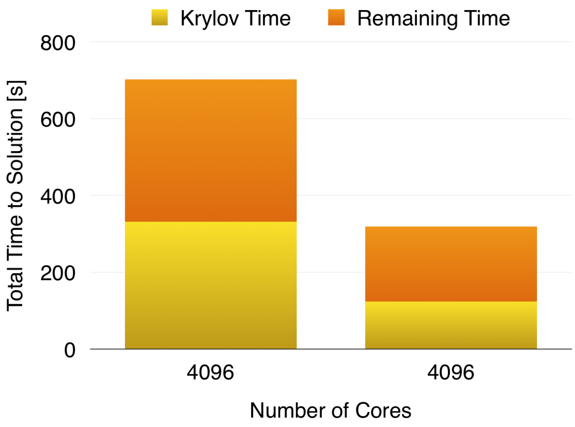

Fig. 3.4: Comparison of exact Nonlinear-FETI-DP-2, cf. right column, with the more traditional Newton-Krylov-FETI-DP approach, cf. left column, on a Cray XT6 (Universität Duisburg-Essen). In both methods a line search method with a Wolfe condition is used as globalization technique. The total time to solution can be reduced by a factor of two for this problem; published in [78].

This results verify the similar results presented in Table 2.9 using exact nonlinear FETI-DP methods and piecewise linear finite elements.

For numerically harder nonlinear problems the time used in the Krylov iteration can constitute a larger portion of the time to solution. Recall results for one such example are given in Fig. 3.4. The model problem considered here is the third benchmark problem defined in Section 2.7.2: We have considered the p-Laplace with $p = 4$ and a multiplicative weight of $\alpha = 10^6$ in 64 channels of width $1/4H$ and $p = 2, \alpha = 1$, elsewhere. In this example the total time to solution is reduced by a factor of two by using the nonlinear method already on 4096 cores. Here we used our exact FETI-DP implementations. For a comparison of the different methods we also recall the results from Table 2.9 and the corresponding discussion.

3.4.4 Weak Parallel Scalability

Weak Parallel Scalability on the Mira BlueGene/Q (Argonne National Laboratory)

In Table 3.2 and Fig. 3.5 weak parallel scalability results on the Mira BlueGene/Q node are presented for the p-Laplace $(-(\Delta + 4\Delta_p)u = 1, p = 4, H/h = 180)$. Again, the total time to solution, denoted total execution time in the table, includes the time for the assembly of the problem as well as its solution. For the parallel efficiency, our reference base line is the time on a single node (16

Vulcan BlueGene/Q Supercomputer (LLNL)							
Cores	Problem Size	Solver	Newton Steps Phase1/2	Krylov Iter	Krylov Time	Total Execution Time	Parallel Effic.
16	1 050 625	NK	– / 20	219	57.5s	**459.7s**	100%
		NL	13 / 7	84	20.3s	**332.0s**	100%
64	4 198 401	NK	– / 21	438	110.4s	**534.9s**	86%
		NL	14 / 7	138	33.17s	**356.1s**	93%
256	16 785 409	NK	– / 22	546	137.9s	**587.1s**	78%
		NL	14 / 8	185	44.94s	**396.6s**	84%
1 024	67 125 249	NK	– / 23	599	152.7s	**624.3s**	74%
		NL	15 / 8	208	51.3s	**413.6s**	80%
4 096	268 468 225	NK	– / 23	624	160.6s	**630.1s**	73%
		NL	16 / 7	199	49.6s	**410.6s**	81%
16 384	1 073 807 361	NK	– / 24	688	178.4s	**676.9s**	68%
		NL	16 / 8	228	57.5s	**441.2s**	75%
65 536	4 295 098 369	NK	– / 24	722	190.0s	**720.5s**	64%
		NL	17 / 7	211	54.1s	**453.9s**	73%

Table 3.1: New, irNonlinear-FETI-DP-1 (abbreviated by NL) algorithm compared to more traditional Newton-Krylov irFETI-DP (abbreviated by NK) on the Vulcan supercomputer at Lawrence Livermore National Laboratory; p-Laplace inclusions in linear Laplace equation; $p = 4$; $H/h = 128$; piecewise quadratic finite elements. The base line for the parallel efficiency is a single BlueGene/Q node (16 cores); published in [78].

cores). The parallel scalability is satisfactory, although we do see an increase in the total time especially when scaling from 65K to 262K cores. A parallel efficiency exceeding 100% stems from a variation in the number of Newton steps by one iteration. In Table 3.3 the weak parallel scalability for both solution phases is shown separately. As expected, since it includes significantly less communication, Phase 1 scales better than Phase 2.

In the next paragraphs, we discuss possible reasons for some of the inefficiencies observed and we are able to present improved results on Vulcan after having implemented some modifications.

Analysis and Discussion of the Results on the Mira BlueGene/Q

A better understanding why the time to solution increases when scaling from 65K to 262K cores on Mira can be achieved by an analysis of several subtimers. Considering Fig. 3.6 we can see that a better scalability for Phase 1 is prevented by an noticeable, undesired increase of the time used in the Krylov iterations

Mira BlueGene/Q Supercomputer (ANL)						
					Total	
	Problem	**Newton Steps**	**Krylov**		**Execution**	**Parallel**
Cores	**Size**	**Phase 1/Phase 2**	**Iter**	**Time**	**Time**	**Effic.**
16	514 089	5 / 1	10	1.23s	64.4s	100%
64	2 053 489	5 / 1	18	2.16s	66.0s	98%
256	8 208 225	4 / 1	23	3.17s	57.0s	113%
1 024	32 821 441	4 / 1	23	3.79s	57.9s	111%
4 096	131 262 849	4 / 1	22	4.04s	58.2s	111%
16 384	525 005 569	5 / 1	21	3.97s	69.8s	92%
65 536	2 099 930 625	4 / 1	23	3.59s	64.9s	99%
262 144	8 399 539 201	4 / 1	23	4.77s	90.2s	71%

Table 3.2: Weak scalability for the irNonlinear-FETI-DP-1 algorithm with Wolfe line search on the Mira supercomputer at Argonne National Laboratory; see also the detailed analysis in Fig. 3.6. Improvements resulting from this analysis are discussed in the text; published in [78].

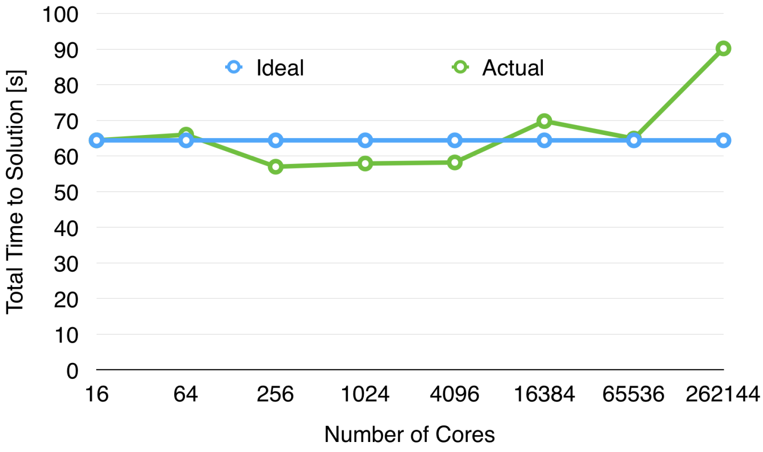

Fig. 3.5: Weak parallel scalability on the Mira BlueGene/Q; cf. the data in Table 3.2; published in [78].

(yellow color in Fig. 3.6). Since, in Phase 1, the Krylov method operates exclusively on the coarse problem, this suggests an inefficiency in the coarse problem. Indeed, for this problem, as a result of the larger number of Newton iterations in Phase 1, the coarse problem has to be solved more often in Phase 1 than in Phase 2. For improving scalability, we therefore have to revisit the coarse problem. In fact, in recent results on the Vulcan BlueGene/Q the scalability of the coarse problem solve has been improved by optimizing several BoomerAMG

Mira BlueGene/Q Supercomputer (ANL)					
		Phase 1		Phase 2	
	Problem	Newton step		Newton step	
Cores	Size	ø Time	Effic.	ø Time	Effic.
16	514 089	10.0s	100%	14.4s	100%
64	2 053 498	10.1s	99%	15.5s	93%
256	8 208 225	10.1s	99%	16.5s	87%
1 024	32 821 441	10.2s	98%	17.1s	84%
4 096	131 262 849	10.2s	98%	17.4s	83%
16 384	525 005 569	10.4s	96%	17.7s	81%
65 536	2 099 930 625	11.6s	86%	18.7s	77%
262 144	8 399 539 201	16.0s	63%	26.1s	55%

Table 3.3: Scalability of the two different solution phases for irNonlinear-FETI-DP-1 algorithm with Wolfe line search on the Mira supercomputer at Argonne National Laboratory; see also the detailed analysis in Fig. 3.6. Improvements resulting from this analysis are discussed in the text; published in [78].

parameters and by using a better distribution of the coarse problem to MPI ranks; see Fig. 3.7, the following paragraphs, and Section 3.4.5.

Second, there is a noticeable increase in the remaining time phase (orange color in Fig. 3.6), especially in Phase 2 of the algorithm. Here, the remaining time consists of the update of the solution, the computation of some parallel norms for the global Newton iteration, and the cleanup of some parallel data structures. Further investigations on the Vulcan BlueGene/Q, see Section 3.4.4, have exposed an inefficiency in the parallel update of the Lagrange multipliers which affected the solution phases of all our nonlinear FETI-DP implementations. The update was then replaced by an improved implementation which subsequently was used to obtain the weak scaling results for Vulcan in Table 3.8 and Fig. 3.10 and the strong scaling results presented in Table 3.9 and Fig. 3.11.

Third, for 262K cores we see an unexpected increase in the portion of the total time that is spent in the FETI-DP setup (green color in Fig. 3.6). This phase contains the assembly of the coarse problem, the local LU factorizations, the construction of several scatters, the AMG setup and the construction of the Dirichlet preconditioner (only necessary in Phase 2). The increase in runtime of the FETI-DP setup time affects both, Phase 1 and Phase 2. Since we expect the local factorizations to scale perfectly the increase can only be a result of the coarse setup, i.e., the construction of the coarse problem; cf. Section 3.3.2. This has been confirmed by a detailed additional analysis on the Vulcan Blue-

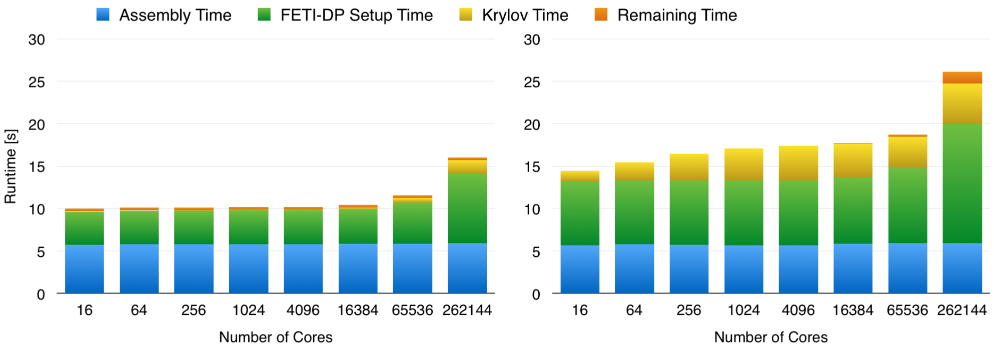

Fig. 3.6: Detailed analysis of Phase 1 (left) and Phase 2 (right) performed on the Mira supercomputer of the first implementation of the irNonlinear-FETI-DP-1 algorithm; cf. Table 3.3. Improvements resulting from this analysis are discussed in the text; see also the improved results on Vulcan in Fig. 3.10; published in [78].

Gene/Q; see the following paragraphs for the discussion. Better results have then been achieved on Vulcan by avoiding a certain redistribution step for the coarse problem. We can conclude that not only the solution of the FETI-DP coarse problem but also its efficient construction will be the key to obtain scalability to one million cores.

The results presented in this chapter are based on pure MPI, i.e., the largest problem uses 262K MPI ranks. A hybrid MPI/OpenMP model offers further potential to increase scalability. We can reduce the number of MPI ranks by using a threaded solver on the subdomain level. Results for this approach are promising but beyond the scope of this thesis. Hybrid MPI/OpenMP aspects and corresponding computational results have been discussed elsewhere [80]. Threaded BLAS or LAPACK libraries were also not used here. From our experience on BlueGene/P, the sparse direct solver [28] does not profit from the fine grained parallelism of a threaded BLAS.

Weak Parallel Scalability on the SuperMUC Sandy Bridge Petascale System (Leibniz Supercomputing Centre, Munich)

The same model problem as on Mira has been solved on SuperMUC at Leibniz Rechenzentrum (LRZ) in Munich, Germany. The weak parallel scalability for both phases can be seen in Table 3.4, showing a similar behavior as on Mira. For the nonlinear hyperelasticity problem, we have similar results but on SuperMUC so far on a smaller scale; see Table 3.5.

SuperMUC (Leibniz Supercomputing Centre)							
		Phase 1		**Phase 2**			
	Problem	**Newton step**		**Newton step**		**Total**	
Cores	**Size**	**ø Time**	**Effic.**	**ø Time**	**Effic.**	**Time**	**Effic.**
32	9M	11.1s	100%	16.2s	100%	112.5s	100%
128	38M	11.4s	97%	17.2s	94%	117.8s	96%
512	151M	11.4s	97%	17.2s	94%	119.1s	95%
2 048	604M	11.5s	97%	17.4s	93%	119.1s	95%
8 192	2 416M	12.3s	90%	18.8s	86%	127.9s	88%
32 768	9 664M	14.6s	76%	23.3s	70%	151.4s	74%

Table 3.4: Inexact reduced Nonlinear-FETI-DP-1 algorithm on the SuperMUC supercomputer at Leibniz-Rechenzentrum in Munich; $-\Delta u - 4\Delta_p u = 1, p = 4, Hx/hx = 768, Hy/hy = 384$; published in [78].

SuperMUC (Leibniz Supercomputing Centre)							
		Phase 1		**Phase 2**			
	Problem	**Newton step**		**Newton step**		**Total**	
Cores	**Size**	**ø Time**	**Effic.**	**ø Time**	**Effic.**	**Time**	**Effic.**
64	4M	15.9s	100%	26.2s	100%	126.6s	100%
256	16M	17.1s	93%	29.1s	90%	138.5s	91%
1024	67M	17.5s	91%	30.9s	85%	127.9s	99%
4096	268M	19.6s	81%	34.2s	77%	141.8s	89%

Table 3.5: Inexact reduced Nonlinear-FETI-DP-1 algorithm on the SuperMUC supercomputer at Leibniz-Rechenzentrum in Munich; Neo-Hooke material; $E = 210000$ in circular inclusions and $E = 210$ in the surrounding matrix material; Poisson ratio $\nu = 0.3$ in the complete domain; see Fig. 3.3 for the geometry; a fixed displacement of 1% in x-direction is prescribed in each boundary node; published in [78].

Weak Parallel Scalability on the Vulcan BlueGene/Q (Lawrence Livermore National Laboratory)

In Table 3.6 the weak scalability of the complete application for a nonlinear and two-dimensional elasticity problem is presented. Here, we consider a rectangular domain $\Omega = (0,0) \times (2,1)$ and apply a fixed 1% displacement in x-direction in each degree of freedom on the boundary, i.e.

$$x = \begin{bmatrix} 1.01 & 0 \\ 0 & 1 \end{bmatrix} X$$

for all nodes $X \in \partial\Omega$. We use a Neo-Hooke material model and consider a heterogenous material characterized by two different elasticity modules; cf. Section 3.4.1 and Fig. 3.3 for the distribution of the coefficients. The smallest problem with 1.6 million degrees of freedom is decomposed into 32 subdomains and the largest problem with 6.7 billion degrees of freedom is decomposed into 131 072 subdomains. All in all, a nearly constant behavior of the time spent in Krylov iterations can be observed. This is comparable to the behavior of our irNonlinear-FETI-DP-1 method applied to the p-Laplace model problem; cf. Section 3.4.3. The parallel efficiency of the application is always better than 90%. But again we see an increase in runtime and, consequently, a drop in efficiency from 96% to 90% when scaling from 32K to 131K cores. We provide detailed time measurements in the next paragraph to explain this effect, and how we obtain improvements.

We also present weak scaling results for nonlinear elasticity in three dimensions; see Table 3.7. The problem setup in this experiment is similar to the two-dimensional model problem: We have one stiff spherical inclusion ($E = 210000$) in each subdomain embedded in softer material ($E = 210$). To increase the memory available for each subdomain, we use 8 of the 16 cores of one Vulcan node. The parallel efficiency on 65K cores compared to 128 cores is 78% and thus seems satisfactory.

Vulcan BlueGene/Q Supercomputer (LLNL)						
					Total	
	Problem	**Newton Steps**	**Krylov**		**Execution**	**Parallel**
Cores	**Size**	**Phase 1/Phase 2**	**Iter**	**Time**	**Time**	**Effic.**
32	1 642 242	4 / 3	93	23.2s	**250.3s**	100%
128	6 561 282	4 / 3	107	27.2s	**256.0s**	98%
512	26 229 762	4 / 3	109	27.9s	**257.7s**	97%
2 048	104 888 322	4 / 3	109	28.4s	**258.7s**	97%
8 192	419 491 842	4 / 3	107	28.7s	**261.0s**	96%
32 768	1 677 844 482	4 / 3	105	27.2s	**261.7s**	96%
131 072	6 711 132 162	4 / 3	102	26.7s	**278.9s**	90%

Table 3.6: Inexact reduced Nonlinear-FETI-DP-1 algorithm on the Vulcan supercomputer at Lawrence Livermore National Laboratory; Neo-Hooke material; $E = 210000$ in circular inclusions and $E = 210$ in the surrounding matrix material; Poisson's ratio $\nu = 0.3$ in the complete domain; see Fig. 3.3 for the geometry; a fixed displacement of 1% in x-direction is prescribed in each boundary node; two-dimensional and rectangular domain $(0,0) \times (2,1)$; H/h=80; piecewise quadratic finite elements; published in [78].

Vulcan BlueGene/Q Supercomputer (LLNL)							
		Krylov			**Average**	**Total**	
	Problem			**Newton**	**Time**	**Execution**	**Parallel**
Cores	**Size**	**Iter**	**Time**	**Steps**	**Newton Step**	**Time**	**Effic.**
128	698 691	85	9.3s	4	22.2s	**88.7s**	100%
1 024	5 447 811	98	11.8s	4	23.1s	**92.2s**	96%
8 192	43 022 595	116	15.5s	4	24.7s	**98.9s**	90%
65 536	341 955 075	181	27.2s	4	28.5s	**113.9s**	78%

Table 3.7: Newton-Krylov irFETI-DP on the Vulcan supercomputer at Lawrence Livermore National Laboratory; Neo-Hooke material; $E = 210000$ in spherical inclusions and $E = 210$ in the surrounding matrix material; Poisson's ratio $\nu = 0.3$ in the complete domain; a fixed displacement of 1% in x-direction is prescribed in each boundary node; three-dimensional cubic domain; H/h=7; piecewise quadratic finite elements; published in [78].

Analysis and Discussion of the Results on the Vulcan BlueGene/Q

In Fig. 3.7, we partition the average runtime of one Newton step in four different phases: Assembly of the local problems (blue color in Fig. 3.7), FETI-DP and preconditioner setup including the local LU factorizations (green color in Fig. 3.7), Krylov iteration (yellow color in Fig. 3.7), and the remaining time (orange color in Fig. 3.7). In Fig. 3.7 (left) we present the average timings over all Newton steps of Phase 1, in Fig. 3.7 (right) the respective timings for Phase 2 (solution phase). Let us remark that Phase 2 is comparable to classical Newton-Krylov irFETI-DP.

The assembly time (blue color in Fig. 3.7) scales, as one should expect, nearly perfectly for both phases.

In Phase 1, the Krylov iteration time can now be neglected after our improvements on the coarse problem solve: We have better adapted BoomerAMG to the properties of our FETI-DP coarse operator and use a better distribution of the coarse problem to the MPI ranks, i.e., we use, e.g., 16K cores for the coarse problem in the run using a total of 131K cores. The combination of these two specific changes alone results in a noticeable improvement compared to the Mira results.

In Phase 2, the time spent in the Krylov method takes a larger part, but scales quite well, also because the nonlinear FETI-DP method keeps the number of Krylov iterations nearly constant. We also notice that the remaining time (orange color in Fig. 3.7) does not scale perfectly when reaching 131K cores. This was caused by an inefficiency in the implementation of the update of the Lagrange multipliers, already mentioned in the discussion of the Mira results. This does not affect this weak scaling experiment on Vulcan substantially, but proved a crucial point in later strong scaling tests on Vulcan. We have reimplemented the update and present corresponding results in Section 3.4.5.

A noticeable parallel inefficiency in the FETI-DP setup time (green color in Fig. 3.7) is caused by the assembly of the coarse problem. This corresponds to the earlier results on Mira. We present a detailed analysis of the FETI-DP setup time in Phase 1 in Fig. 3.8. The FETI-DP setup phase is split into the LU factorizations (green color in Fig. 3.8), the assembly and redistribution of the globally coupled coarse problem (red and purple colors in Fig. 3.8), the construction of the scatters representing B and R_Π (yellow color in Fig. 3.8), and the BoomerAMG setup (orange color in Fig. 3.8). Apparently, only the phases related to the coarse problem are not scaling perfectly. It is possible to avoid the redistribution process, see Section 3.3.2, by assembling the coarse problem directly on a subset of MPI ranks. This proves to be more efficient:

We reimplemented this part of the code and repeated the tests from Table 3.6. The improved results can be found in Table 3.8 and Fig. 3.9. In Fig. 3.9 the redistribution (purple color in Fig. 3.9 and Fig. 3.8) is removed resulting in an improvement of the overall parallel efficiency from 90% (Table 3.6) to 92% in Table 3.8 and Fig. 3.10.

Nevertheless, we assume that improvements are still possible in the construction of the coarse problem since a detailed performance analysis for a number of cores this large has not previously been undertaken for this code. We may be able to learn from highly scalable parallel multigrid codes which also rely on an efficient construction of the coarse operator.

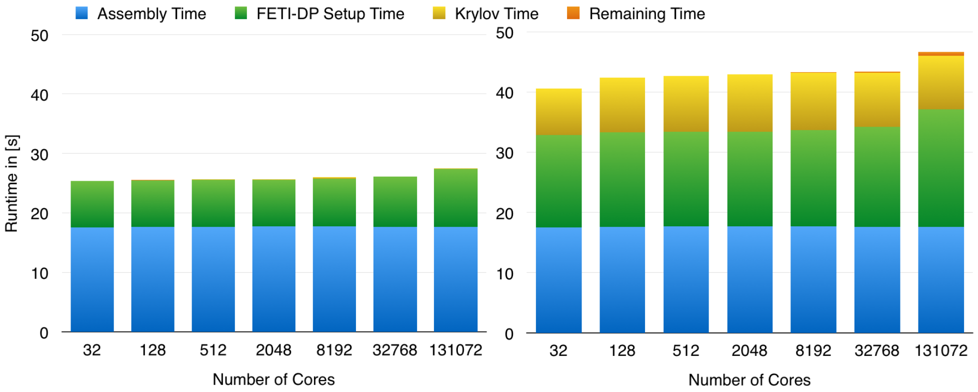

Fig. 3.7: Detailed analysis of Phase 1 (left) and Phase 2 (right) of the irNonlinear-FETI-DP-1 method; average time per Newton step; corresponds to the data in Table 3.6; published in [78].

Vulcan BlueGene/Q Supercomputer (LLNL)							
Cores	**Problem Size**	**Newton Steps Phase 1/Phase 2**	Krylov **Iter**	**Time**	**Total Execution Time**	**Parallel Effic.**
32	1 642 242	4 / 3	93	23.2s	**249.9s**	100%
128	6 561 282	4 / 3	107	27.4s	**256.2s**	98%
512	26 229 762	4 / 3	108	27.9s	**258.2s**	97%
2 048	104 888 322	4 / 3	109	28.6s	**258.9s**	97%
8 192	419 491 842	4 / 3	106	28.9s	**261.2s**	96%
32 768	1 677 844 482	4 / 3	105	28.3s	**263.7s**	95%
131 072	6 711 132 162	4 / 3	102	26.8s	**273.1s**	92%

Table 3.8: Setting as in Table 3.6; assembly of the coarse problem on a subset of $\min(\text{MPI-size}, 16K)$ cores instead of redistribution of the coarse space; published in [78].

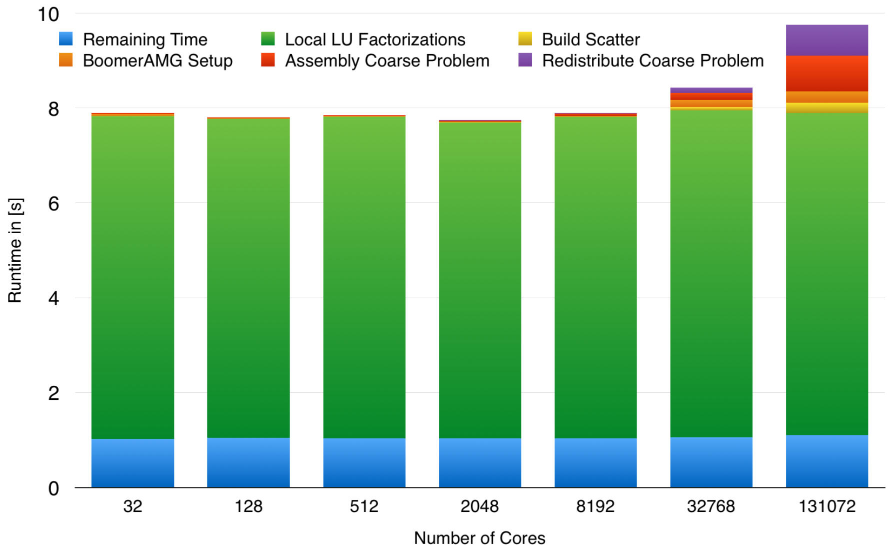

Fig. 3.8: Detailed subtimers of the irFETI-DP setup in Phase 1; average time per Newton step; corresponds to the data in Table 3.6; published in [78].

.

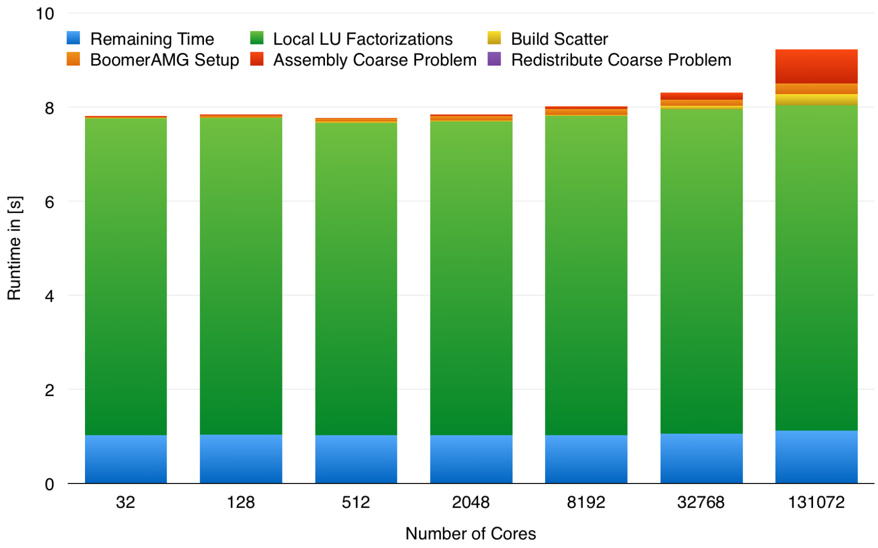

Fig. 3.9: Detailed subtimers of the irFETI-DP setup in Phase 1; average time per Newton step; corresponds to the data in Table 3.8. Improved results compared to Fig. 3.8; published in [78].

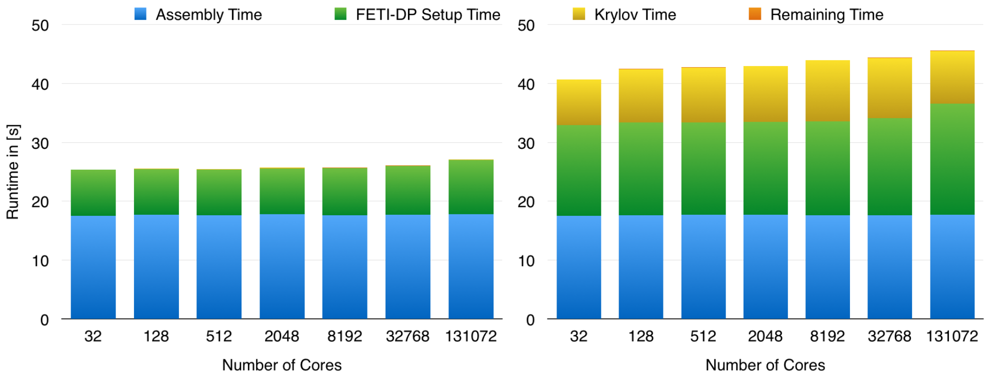

Fig. 3.10: Detailed analysis of Phase 1 (left) and Phase 2 (right) of the hybrid irNonlinear-FETI-DP-1 method; average time per Newton step; corresponds to the data in Table 3.8. Improved results compared to Fig. 3.7; published in [78].

3.4.5 Strong Parallel Scalability

Finally, we present strong scaling tests for a nonlinear elasticity problem in two dimensions; see Table 3.9. We, again, use the problem setup with stiff circular inclusions, cf. Fig. 3.3. We decompose a problem with 419 million degrees of freedom and 131 072 inclusions into 131 072 subdomains and solve with our new inexact reduced Nonlinear-FETI-DP-1 method using an increasing numbers of BlueGene/Q cores. We start with 1 024 cores (16 nodes) and scale up to 131K cores (8192 nodes). This implies that we solve between one and 128 subdomain problems on each processor core, and each of these has approximately 6.6K degrees of freedom, which is comparable small. We are not able to start our strong scaling test on fewer cores, due to memory constraints. The result of 63% parallel efficiency on 131K cores is convincing. Let us also remark that we even gain a mentionable speedup when scaling from 65K to 131K cores. A graphical presentation of the parallel speedup can be found in Fig. 3.11.

We also provide and discuss some detailed timings for this strong scaling experiment. In Fig. 3.12, we present four different subtimers: Krylov iterations (yellow color in Fig. 3.12), assembly of local problems (blue color in Fig. 3.12), FETI-DP setup (green color in Fig. 3.12) and updates and parallel norms (orange color in Fig. 3.12). On the left, the average measurements of these subtimers for each Newton step of Phase 2 is presented. On the right, the percentage as portion of the total time for these subtimers is depicted.

Let us remark that three main improvements compared to the Mira results enabled these strong scaling results. First, the elimination of the inefficiency in the update of the Lagrange multipliers. It has made the remaining time (orange color in Fig. 3.12) insignificant. Second, we have removed the redistribution of the coarse problem and replaced it by a faster assembly process on a subset of MPI ranks; see also Section 3.4.4. For the strong scaling results, we assemble the coarse problem at most on 16K cores. This also results in a better scalability of the FETI-DP setup phase. Third, we have better adapted BoomerAMG to the properties of the FETI-DP coarse problem.

Still, the FETI-DP setup phase does not show optimal parallel efficiency and the percentage of this phase of a single Newton step grows (green color in Fig. 3.12). Further investigations and optimizations of the assembly process of the coarse problem may be necessary.

Vulcan BlueGene/Q Supercomputer (LLNL)						
Cores	**Subdomains**	**Problem Size**	**Total Execution Time**	**Actual Speedup**	**Ideal Speedup**	**Parallel Effic.**
1 024	131 072	419 471 361	3 365.1s	1.0	1	100%
2 048	131 072	419 471 361	1 726.4s	1.9	2	97%
4 096	131 072	419 471 361	868.0s	3.9	4	97%
8 192	131 072	419 471 361	453.5s	7.4	8	93%
16 384	131 072	419 471 361	231.4s	14.6	16	91%
32 768	131 072	419 471 361	119.8s	28.1	32	88%
65 536	131 072	419 471 361	64.3s	51.6	64	81%
131 072	131 072	419 471 361	41.7s	80.6	128	63%

Table 3.9: Inexact reduced Nonlinear-FETI-DP-1 algorithm on the Vulcan supercomputer at Lawrence Livermore National Laboratory; Neo-Hooke material; $E = 210000$ in circular inclusions and $E = 210$ in the surrounding matrix material; Poisson's ratio $\nu = 0.3$ in the complete domain; see Fig. 3.3 for the geometry; a fixed displacement of 1% in x-direction is prescribed in each boundary node; piecewise quadratic finite elements; published in [78].

3.4.6 Using Multiple MPI Processes per Core

Each BlueGene/Q core has a maximum number of 4 hardware threads and it is possible to execute up to four MPI processes on each core by specifying the "- -overcommit" option. In strong scaling experiments, as presented above, several subdomain problems have to be solved on each core. It is natural to handle these subdomains sequentially on each core. Alternatively, we can handle subdomains in parallel by using several MPI processes per core, which may help to fill the pipelines and better utilize the execution units. Of course, the memory of the node will be partitioned accordingly. We present results for a Neo-Hooke hyperelasticity problem with 838 942 722 degrees of freedom in Fig. 3.13 and compare computations on 1 024, 2 048, and 4 096 nodes with one process per core (upper, blue line) and four, two, and one MPI processes per core (lower, green line). We use the maximum number of four processes per core on 1 024 nodes and two processes per core on 2 048 nodes. Of course we can not expect perfect scalability, but we do achieve a significant speedup of 30% compared to the standard approach by using two MPI processes per processor core. We then do not obtain a significant additional benefit from using four MPI processes per core. Thus, if feasible an overcommit should be considered.

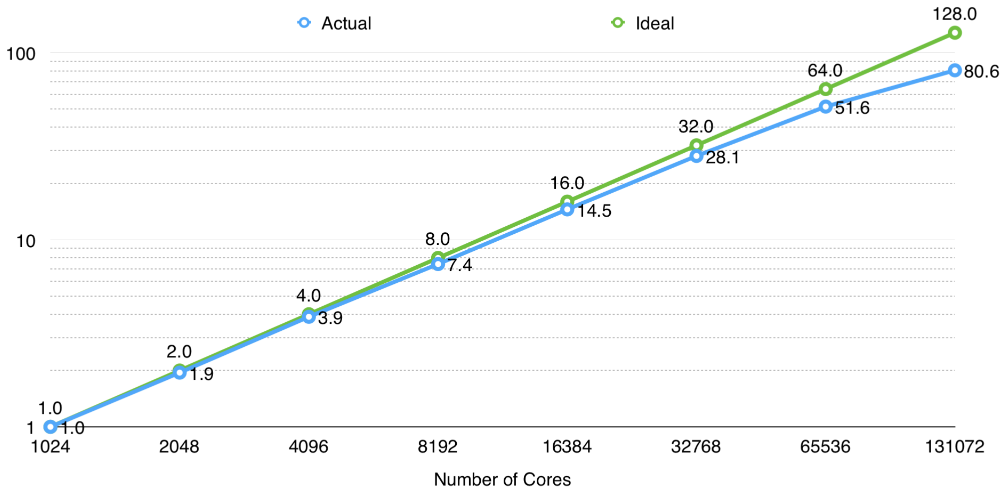

Fig. 3.11: Strong Scaling on Vulcan: Visualization of the speedup from Table 3.9; published in [78].

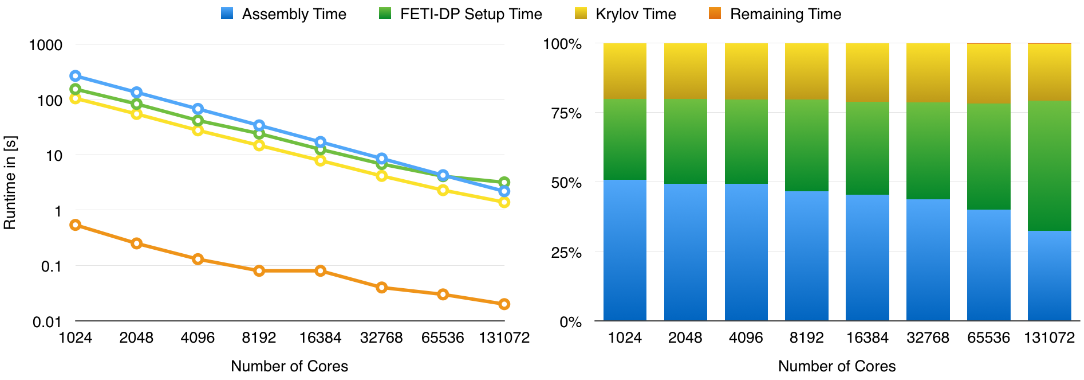

Fig. 3.12: Detailed subtimer for the strong scaling experiments from Table 3.9. Average time per Newton step in Phase 2; published in [78].

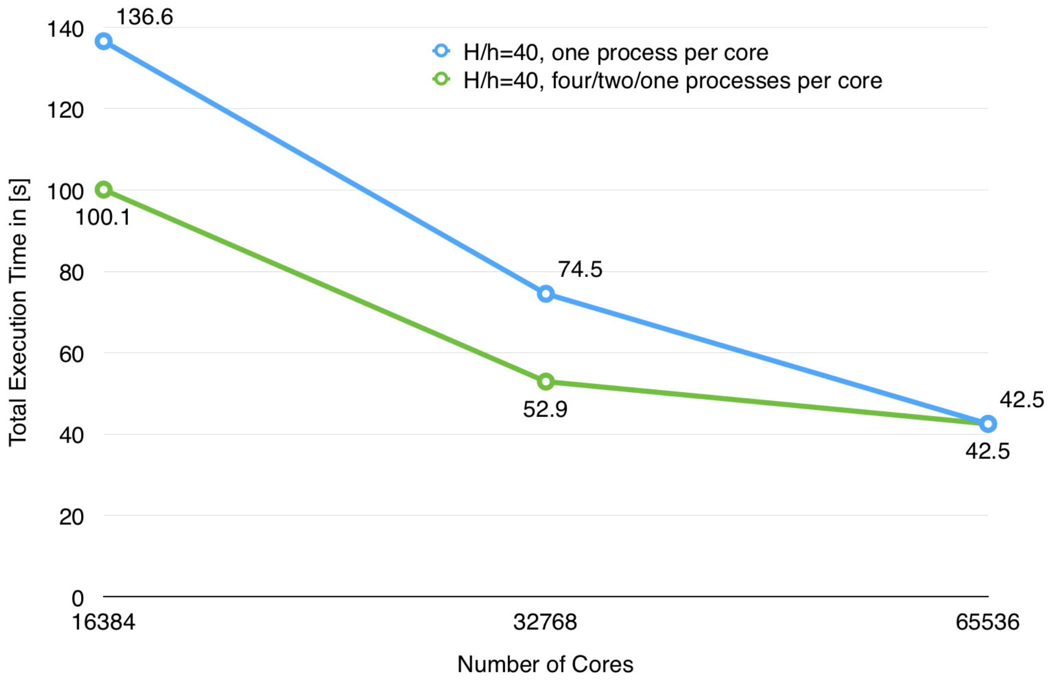

Fig. 3.13: Effect of using multiple MPI processes for each processor core (overcommit): One process per core compared to four/two/one processes per core. For the upper (blue) line we use 65 536 subdomains and 16 384, 32 768, and 65 536 MPI ranks on 16 384, 32 768, and 65 536 BG/Q cores, respectively. For the lower (green) line we use 65 536 subdomains and 65 536 MPI ranks on 16 384, 32 768, and 65 536 BG/Q cores; published in [78].

3.5 Outlook on a New Method: Inexact Reduced Nonlinear-FETI-DP-2

Especially in problems with strongly localized nonlinearities, Nonlinear-FETI-DP-2 often shows the best performance, since localized nonlinear problems are solved in each Newton step on the right hand side; see Chapter 2.7 for numerical results. Unfortunately, the idea of inexact reduced FETI-DP cannot be transferred to Nonlinear-FETI-DP-2 in a straightforward fashion, since we perform a nonlinear elimination of our coarse space variables $\tilde{u}_\Pi$. Therefore, we present a third nonlinear FETI-DP approach, where a nonlinear elimination of all interior and dual variables $B = [I\;\Delta]$ is performed before linearization. In other words, we suggest to nonlinearly eliminate as many variables as possible before linearization without loosing the opportunity to solve the coarse problem inexactly. We denote this algorithm by inexact reduced Nonlinear-FETI-DP-2 (irNonlinear-FETI-DP-2); see [91].

3.5.1 Derivation of the Method

A Nonlinear Schur Complement

We start again with the nonlinear FETI-DP master system introduced in equation (2.34)

$$\begin{aligned} \widetilde{K}(\tilde{u}) + B^T\lambda &= \tilde{f} \\ B\tilde{u} &= 0. \end{aligned} \tag{3.17}$$

Recalling the partitioning $\tilde{u} := \left[u_B^T, \tilde{u}_\Pi^T\right]^T$ and $\tilde{f} := \left[f_B^T, \tilde{f}_\Pi^T\right]^T$ into primal variables Π and interior and interface variables $B = [I\;\Delta]$, and, accordingly, $\widetilde{K}(\tilde{u}) := \left[K_B(u_B,\tilde{u}_\Pi)^T, \widetilde{K}_\Pi(u_B,\tilde{u}_\Pi)^T\right]^T$, we can reformulate (3.17) to

$$\begin{aligned} K_B(u_B,\tilde{u}_\Pi) + B_B^T\lambda \quad -f_B &= 0 \\ \widetilde{K}_\Pi(u_B,\tilde{u}_\Pi) \quad -\tilde{f}_\Pi &= 0 \\ B_B u_B &= 0. \end{aligned} \tag{3.18}$$

We analogously partition the tangential matrix $D\widetilde{K}(\tilde{u})$ and obtain

$$D\widetilde{K}(\tilde{u}) = \left[\begin{array}{cc} D_{u_B}K_B(u_B,\tilde{u}_\Pi) & D_{\tilde{u}_\Pi}K_B(u_B,\tilde{u}_\Pi) \\ D_{u_B}\widetilde{K}_\Pi(u_B,\tilde{u}_\Pi) & D_{\tilde{u}_\Pi}\widetilde{K}_\Pi(u_B,\tilde{u}_\Pi) \end{array}\right] =: \left[\begin{array}{cc} D\widetilde{K}(\tilde{u})_{BB} & D\widetilde{K}(\tilde{u})_{\Pi B}^T \\ D\widetilde{K}(\tilde{u})_{\Pi B} & D\widetilde{K}(\tilde{u})_{\Pi\Pi} \end{array}\right]. \tag{3.19}$$

Instead of linearizing equation (3.17), which would lead to NL-FETI-DP-1, or instead of performing a nonlinear elimination of $\tilde{u}$ in equation (3.17), which would lead to NL-FETI-DP-2, we perform a nonlinear elimination of u_B in (3.18).

This finally results in concurrent local nonlinear problems in the variable u_B. In order to derive our inexact reduced Nonlinear-FETI-DP-2 method, we first introduce a nonlinear Schur complement on $\left[\tilde{u}_\Pi^T, \lambda^T\right]^T$.
Let us assume that $\left[u_B^{*^T}, \tilde{u}_\Pi^{*^T}, \lambda^{*^T}\right]^T$ is a solution of (3.18). Under the sufficient assumption that K_B is continuously differentiable, that the solution $\left[u_B^{*^T}, \tilde{u}_\Pi^{*^T}, \lambda^{*^T}\right]^T$ exists and satisfies $K_B(u_B^*, \tilde{u}_\Pi^*) + B_B^T \lambda^* - f_B = 0$, and that $D\widetilde{K}(u_B^*, \tilde{u}_\Pi^*)_{BB}$ is invertible, there exists an implicit function $h(\tilde{u}_\Pi^*, \lambda^*) = u_B^*$. Additionally, we then have

$$K_B(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi) + B_B^T \lambda - f_B = 0, \tag{3.20}$$

in a sufficiently small neighborhood of $\left[\tilde{u}_\Pi^{*^T}\ ,\ \lambda^{*^T}\right]^T$. From the implicit function theorem we obtain the derivative of h,

$$Dh(\tilde{u}_\Pi, \lambda) = [D_{\tilde{u}_\Pi} h(\tilde{u}_\Pi, \lambda) \quad D_\lambda h(\tilde{u}_\Pi, \lambda)] \tag{3.21}$$

with

$$D_{\tilde{u}_\Pi} h(\tilde{u}_\Pi, \lambda) = -(D\widetilde{K}(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi))_{BB}^{-1}\, D\widetilde{K}(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi)_{\Pi B}^T \tag{3.22}$$

and

$$D_\lambda h(\tilde{u}_\Pi, \lambda) = -(D\widetilde{K}(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi))_{BB}^{-1}\, B_B^T \tag{3.23}$$

Inserting the implicit function in row two and three of (3.18), we can define a nonlinear Schur complement by

$$S(\tilde{u}_\Pi, \lambda) := \begin{bmatrix} \widetilde{K}_\Pi(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi) - \tilde{f}_\Pi \\ B_B h(\tilde{u}_\Pi, \lambda) \end{bmatrix}. \tag{3.24}$$

irNonlinear-FETI-DP-2 is then linearizing the Schur complement problem $S(\tilde{u}_\Pi^*, \lambda^*) = 0$ with Newton's method. We obtain the iteration

$$\begin{bmatrix} \tilde{u}_\Pi^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_\Pi^{(k)} \\ \lambda^{(k)} \end{bmatrix} - (DS(\tilde{u}_\Pi^{(k)}, \lambda^{(k)}))^{-1} S(\tilde{u}_\Pi^{(k)}, \lambda^{(k)}). \tag{3.25}$$

Using (3.22), (3.23), and the notation from (3.19) leads to

$$DS(\tilde{u}_\Pi, \lambda) = \tag{3.26}$$

$$\begin{bmatrix} (D\widetilde{K}(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi))_{\Pi\Pi} + (D\widetilde{K}(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi))_{\Pi B} D_{\tilde{u}_\Pi} h(\tilde{u}_\Pi, \lambda) & (D\widetilde{K}(h(\tilde{u}_\Pi, \lambda), \tilde{u}_\Pi))_{\Pi B} D_\lambda h(\tilde{u}_\Pi, \lambda) \\ B_B D_{\tilde{u}_\Pi} h(\tilde{u}_\Pi, \lambda) & B_B D_\lambda h(\tilde{u}_\Pi, \lambda) \end{bmatrix}$$

$$= \begin{bmatrix} (D\widetilde{K})_{\Pi\Pi} - (D\widetilde{K})_{\Pi B}((D\widetilde{K})_{BB})^{-1}(D\widetilde{K})_{B\Pi} & -(D\widetilde{K})_{\Pi B}((D\widetilde{K})_{BB})^{-1}B_B^T \\ -B_B((D\widetilde{K})_{BB})^{-1}(D\widetilde{K})_{B\Pi} & -B_B((D\widetilde{K})_{BB})^{-1}B_B^T \end{bmatrix}$$

$$\overset{(3.4)}{=} \mathscr{A}_r(\tilde{u}_\Pi, \lambda). \qquad (3.27)$$

Here, we partially removed variables $h(\tilde{u}_\Pi, \lambda)$ and $\tilde{u}_\Pi$ for simplicity. If performed in the same linearization point $\tilde{u}$, the derivative of the nonlinear Schur complement in (3.26) is identical to the left hand side $\mathscr{A}_r$ in equation (3.4) of the inexact reduced Nonlinear-FETI-DP-1 method. Therefore, we can again solve the linearized system from (3.25) using an Krylov space method and the preconditioner defined in (3.8); see Section 3.1 for details.

A Local Nonlinear Problem

To obtain the linearization point

$$\left[u_B^{(k)^T}, \tilde{u}_\Pi^{(k)^T}\right]^T := \left[h(\tilde{u}_\Pi^{(k)}, \lambda^{(k)})^T, \tilde{u}_\Pi^{(k)^T}\right]^T$$

in (3.25), we have to solve the local nonlinear problem given in the first row of (3.18)

$$K_B(h(\tilde{u}_\Pi^{(k)}, \lambda^{(k)}), \tilde{u}_\Pi^{(k)}) + B_B\lambda^{(k)} - f_B = 0. \qquad (3.28)$$

Since $\tilde{u}_\Pi^{(k)}$ and $\lambda^{(k)}$ are given as results of the k-th step of the global Newton iteration (3.25), we are only interested in $u_B^{(k)} := h(\tilde{u}_\Pi^{(k)}, \lambda^{(k)})$ and solve equation (3.28) by applying Newton's method. We obtain the local iteration

$$u_B^{(l+1)} = u_B^{(l)} - (D\widetilde{K}(u_B^{(l)}, \tilde{u}_\Pi^{(k)}))_{BB}^{-1}\,(K_B(u_B^{(l)}, \tilde{u}_\Pi^{(k)}) + B_B\lambda^{(k)} - f_B). \qquad (3.29)$$

Regarding the diagonal block structure of $D\widetilde{K}(\cdot)_{BB}^{-1}$, see (3.13), all computations in (3.29) are completely local to the subdomains Ω_i and therefore iteration (3.29) is embarrassingly parallel. Nearest neighbor communication is only needed once for the computation of $B_B\lambda^{(k)}$, since $\lambda^{(k)}$ does not change during the local iteration.

3.5.2 Algorithmic Description

We present the inexact reduced Nonlinear-FETI-DP-2 algorithm in Fig. 3.14. The computational building blocks are identical to the building blocks of inexact reduced Nonlinear-FETI-DP-1 and therefore already described in Section 3.3.

```
Inexact Reduced Nonlinear-FETI-DP-2
    Init: $(u_B^{(0)}, \tilde{u}_\Pi^{(0)}) = \tilde{u}^{(0)} \in \widehat{W}$, $\lambda^{(0)} = 0$

    for $k = 0, ..., convergence$
        for $l = 0, ..., convergence$
            build: $\widetilde{K}(\tilde{u}^{(l)})$ and $D\widetilde{K}(\tilde{u}^{(l)})$
            solve: $(D\tilde{K}(\tilde{u}^{(l)}))_{BB}\delta u_B^{(l)} = K_B(\tilde{u}^{(l)}) + B_B^T\lambda^{(k)} - f_B$    //see eq. (3.29)
            compute: steplength $\alpha^{(l)}$
            update: $\tilde{u}^{(l+1)} := \tilde{u}^{(l)} - \alpha^{(l)}(\delta u_B^{(l)}, 0)^T$    //update $u_B$
        end
        $\tilde{u}^{(k)} := \tilde{u}^{(l+1)}$
        build: $\tilde{K}(\tilde{u}^{(k)})$, $D\tilde{K}(\tilde{u}^{(k)})$, and $M^{-1}$
        iterative Krylov solve for $(\delta\tilde{u}_\Pi^{(k)T}, \delta\lambda^{(k)T})^T$ using left
        preconditioner $\mathscr{B}_{r,L}^{-1}(\widehat{S}_{\Pi\Pi}^{-1}, D\widetilde{K}(\tilde{u})_{BB}^{-1}, M^{-1})$:
        $\mathscr{A}_r(\tilde{u}_\Pi^{(k)}, \lambda^{(k)})(\delta\tilde{u}_\Pi^{(k)}, \delta\lambda^{(k)})^T = (\widetilde{K}_\Pi(\tilde{u}^{(k)}) - \tilde{f}_\Pi, B_B u_B^{(k)})^T$    //see (3.27)
        compute: steplength $\alpha^{(k)}$
        update: $\lambda^{(k+1)} := \lambda^{(k)} - \alpha^{(k)}\delta\lambda^{(k)}$
        update: $\tilde{u}_\Pi^{(k+1)} := \tilde{u}_\Pi^{(k)} - \alpha^{(k)}\delta\tilde{u}_\Pi^{(k)}$
        $\tilde{u}^{(0)} := (u_B^{(l+1)^T}, \tilde{u}_\Pi^{(k+1)^T})^T$    //primal update
        $\lambda^{(0)} := \lambda^{(k+1)}$

    end
```

Fig. 3.14: Pseudocode of the **Inexact Reduced Nonlinear-FETI-DP-2** algorithm. The application of $\widehat{S}_{\Pi\Pi}^{-1}$ will consist of cycles of a parallel AMG method, for $D\widetilde{K}(\tilde{u})_{BB}^{-1}$ uses concurrent forward-backward substitutions of a sparse direct solvers on the subdomains. See also Section 3.3 for the detailed notations.

4 AMG as Preconditioner in Inexact FETI-DP Methods

Fast convergence in nonlinear inexact reduced FETI-DP methods, as introduced in Chapter 3, can only be obtained if a good preconditioner $\widehat{S}_{\Pi\Pi}$ for $\widetilde{S}_{\Pi\Pi}$ is available; this was already mentioned in Section 3.1. In case the preconditioner is spectrally equivalent to $\widetilde{S}_{\Pi\Pi}$, we can expect the same polylogarithmic bound of the condition number of the preconditioned system as in exact FETI-DP methods; cf (2.11). In all experiments presented in Section 3.4 an AMG method with a classical nodal coarsening, see Section 4.2 for details, was used as preconditioner $\widehat{S}_{\Pi\Pi}$ and the performance was quite satisfying. Nevertheless, it is well known that for more sophisticated linear or nonlinear elasticity problems a problem-related and customized AMG setup can improve performance. Classical AMG approaches are designed for scalar partial differential equations and assume that the nullspace of the operator only consists of constant vectors. This assumption does not hold for many systems of PDEs, as, e.g., elasticity problems, where also rotations are part of the nullspace. Several different approaches to handle linear elasticity problems with AMG methods have been suggested in the last decades, as, e.g., smoothed aggregation [16, 136], unsmoothed aggregation [12,20,111,113,114], AMGe [15], element-free AMGe [67], local optimization problems to incorporate the rigid body modes in the interpolation [37], or the global matrix (GM) approach [3].

In this chapter, we provide a brief overview of AMG and describe the GM approach in Section 4.3, which was first introduced in [3]. This method explicitly incorporates given smooth error vectors into the AMG interpolation in order to handle the correction of these errors in the coarse grid correction. The GM approach can help to improve convergence of AMG especially for elasticity problems.

We present parallel results up to 131 072 BlueGene/Q cores and compare the performance of standard AMG and the GM approach as preconditioner in irNonlinear FETI-DP-1.

Finally, we give a short overview of inexact FETI-DP (iFETI-DP) method; introduced in [84]. In this method, no exact elimination of variables is performed at all and the AMG preconditioner is applied to the complete matrix $D\widetilde{K}$ instead of the Schur complement $\widetilde{S}_{\Pi\Pi}$. We present numerical weak scalability results for iFETI-DP using AMG and GM interpolations as a preconditioner.

4.1 Classical Algebraic Multigrid

Let us first give a brief overview of AMG for scalar PDEs and introduce some notations. We consider the linear system $Au = f$ arising from the discretization of a scalar PDE with the exact solution u^*. Algebraic as well as geometric multigrid methods are based on finding a hierarchy of coarser grids and restricting the linear system to each of them. Finally, the system is only solved on the coarsest grid, which is computationally efficient due to the small problem size. The solution on the coarse grid is then interpolated back to the original grid and, additionally, some smoothing steps are applied to correct the error of the interpolated solution. In algebraic multigrid methods the coarsening is performed without any geometrical or mesh related information only considering the entries of the system matrix A. Descriptions of several coarsening strategies as Ruge-Stüben, HMIS or Falgout, and an investigation of their parallel performance can be found in, e.g., [137]. For a given set of grids, an interpolation operator P is constructed to interpolate quantities from one grid to the next finer one and the transposed operator P^T is a common choice as restriction operator.

Let us, for simplicity, only consider the two grid case with one fine and one coarse grid. For an approximate solution u and the exact solution u^* of the system $Au^* = f$ on the fine grid, we have the relationship $Ae = r$ with the error vector defined by $e := u^* - u$, and the residual defined by $r := f - Au$. One AMG V-cycle to correct or update u then reads

```
1) Smooth $\nu_1$ times on: $Au = f$
2) Compute the residual: $r = f - Au$
3) Solve on the coarse grid: $P^T(A)Pe_c = P^T r$
4) Correct u: $u = u + Pe_c$
5) Smooth $\nu_2$ times on: $Au = f$.
```

A full algebraic multigrid V-cycle with more than two levels is depicted in Fig. 4.1. For more details on AMG methods, see, e.g., [122,132]. In the remainder of this chapter we refer to AMG for scalar PDEs as *classical AMG*.

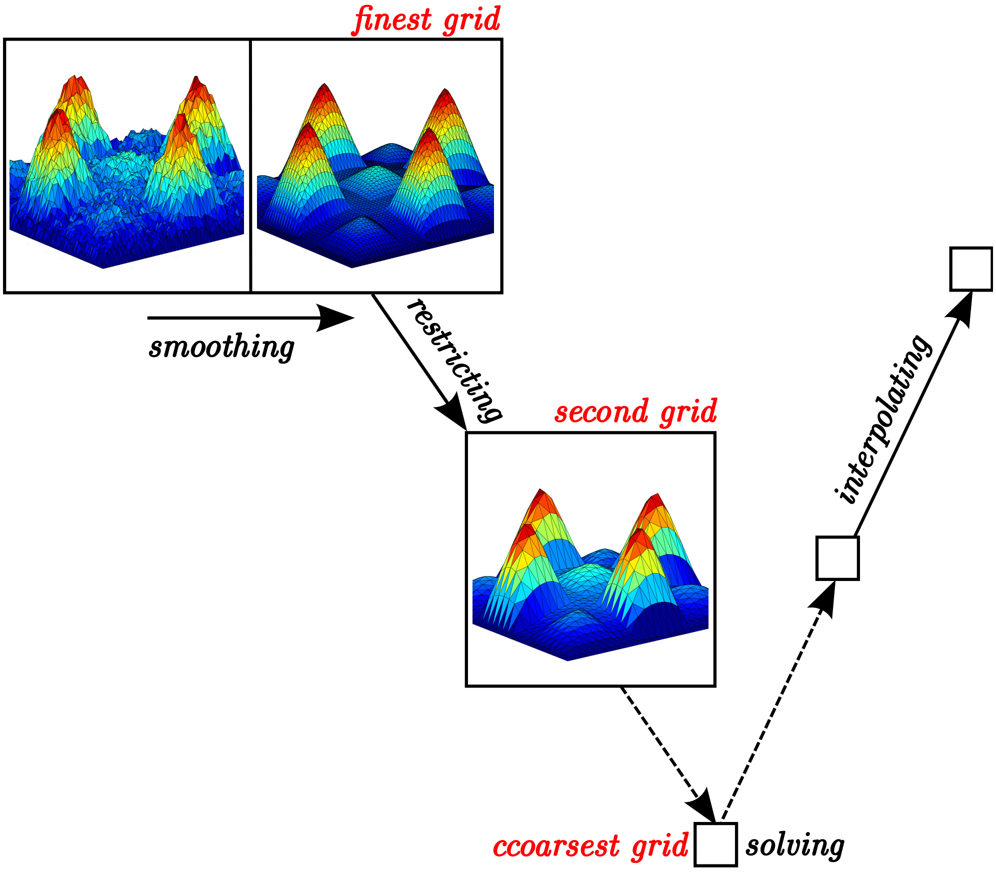

Fig. 4.1: One AMG V-cycle: Smoothing on the fine grid → Restricting to the coarsest grid → Solving on coarsest grid → Interpolating to the finest grid.

4.2 Algebraic Multigrid for Systems of PDEs

We now consider discretized **systems** of partial differential equations $Au = f$. Here, one variable or unknown describes one physical quantity in a grid point or node. For example, in linear or nonlinear elasticity, we have one unknown or degree of freedom describing one spatial direction in each node. For simplicity, we restrict ourselves to the two dimensional case and regard an elasticity problem with two unknowns x and y representing the two spatial directions. A detailed three dimensional description can be found in [3].

In the theory of algebraic multigrid methods there are two common approaches to treat systems of PDEs $Au = f$: the unknown coarsening approach, first described in [122, 132], and the nodal coarsening approach; see, e.g., [24, 62, 121, 122]. While the unknown approach completely separates the different physical quantities, the nodal approach considers all unknowns belonging to the same node at once and thus acts on a nodal basis.

Let us first take a brief look at the unknown approach. Therefore, we assume an unknown related ordering of the system matrix

$$A = \begin{bmatrix} A_{xx} & A_{xy} \\ A_{yx} & A_{yy} \end{bmatrix}. \tag{4.1}$$

The unknown approach is now applying classical AMG coarsening and interpolation strategies to the different variables separately, i.e., applying independently classical AMG to the diagonal blocks A_{xx} and A_{yy}. This strategy apparently neglects all inter-unknown couplings between x and y and the AMG interpolation P has diagonal block structure

$$P = \begin{bmatrix} P_x & 0 \\ 0 & P_y \end{bmatrix}. \tag{4.2}$$

In general, the unknown approach is the most intuitive way to handle systems of PDEs and appears to be effective for problems with a weak coupling between the different unknowns. Of course, performance also strongly depends on the general quality of the chosen coarsening, interpolation, and smoothing techniques with respect to the diagonal problems.

Considering problems with a stronger coupling between the different physical quantities the nodal approach may have a superior performance. Blocking all unknowns sharing the same node together and considering a node related ordering of our system matrix leads to the representation

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} \end{bmatrix}, \tag{4.3}$$

with 2×2 blocks A_{ij} connecting nodes i and j. All in all, we define N as the number of nodes or grid points and thus A is a $N \times N$ block matrix.

Instead of considering the strength of the connection between two variables to construct the coarse grid, we now consider the strength of the connection between two nodes i and j. Therefore, we have to compare the different block entries, as, e.g., A_{ji} or A_{jj}. This is possible by the help of an appropriate measurement as the Frobenius norm $||\cdot||_F$ or the row-sum norm $||\cdot||_\infty$. Applying the chosen norm to the system matrix DK we obtain a condensed $N \times N$ matrix with scalar entries describing the strength of the inter-nodal connections

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1N} \\ c_{21} & c_{22} & \cdots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \dots & c_{NN} \end{bmatrix} := \begin{bmatrix} ||A_{11}|| & ||A_{12}|| & \cdots & ||A_{1N}|| \\ ||A_{21}|| & ||A_{22}|| & \cdots & ||A_{2N}|| \\ \vdots & \vdots & \ddots & \vdots \\ ||A_{N1}|| & ||A_{N2}|| & \dots & ||A_{NN}|| \end{bmatrix}. \tag{4.4}$$

The AMG coarse grids and the interpolation operators can be obtained by applying classical AMG techniques to the condensed matrix C. In the nodal coarsening approach all unknowns in one grid point share the same set of coarse grids. This is not the case in grids obtained from an unknown coarsening, which can lead to completely different coarse meshes for different unknowns.

4.3 The Global Matrix Approach

In general, in algebraic multigrid methods, errors are reduced by two different operations: the smoothing or relaxation steps and the coarse grid correction. For an optimal AMG setup the coarse correction and the relaxation strategy have to be chosen carefully and to complement each other. While simple pointwise relaxation methods as Jacobi or Gauß-Seidel rapidly reduce errors in the directions of eigenvectors associated to large eigenvalues, the reduction in directions of eigenvectors associated to small eigenvalues is less optimal; see [15] for details. Errors, which are only poorly reduced by the smoothing steps are also called *smooth errors* or, more precisely, *algebraic smooth errors* and can be characterized by $Ae \approx 0$ since e is an eigenvector associated to a small eigenvalue. To obtain an effective AMG method these errors should be reduced by the coarse grid correction. For that reason, the smooth error vectors should approximately lie in the range of the interpolation operator P, or, if an error e exactly fulfills the equation $Ae = 0$ on all interior nodes, the error should be interpolated exactly on the fine grid; see, e.g., [15].

In the case of linear elasticity all rotations and translations of the domain (rigid body modes, RBMs) are in the nullspace of the system operator. Since classical AMG interpolations P already interpolate constant vectors exactly, we only have to take care of rotations, or, in two dimensions, the single rotation $s^F(x,y) := [y, -x]$. A possible approach to incorporate an exact interpolation of smooth error vectors in the AMG interpolation is, as already mentioned, the GM approach, first introduced in [3]. We choose the GM approach to optimize convergence of inexact FETI-DP methods for elasticity problems. In the following descriptions we will again restrict ourselves to the case with two grids. A generalization considering the multilevel-case can be found in [3].

The name *global matrix* approach is based on the idea of augmenting a given and global AMG interpolation P with several matrices Q. Each matrix Q has the task to exactly interpolate a chosen smooth error vector. Here, we consider the rotation $s^F := [y, -x]$ in two dimensions as algebraic smooth error. We define s^C as the restriction of s^F onto the coarse grid and augment

$$\widetilde{P} := [P \quad Q], \text{ such that } s^F \in \text{ range}(\widetilde{P}). \tag{4.5}$$

There are several possibilities to define a matrix Q fulfilling equation (4.5) and also retaining the sparsity of P. We define $\widetilde{P}$ in order to fulfill

$$\widetilde{P} \begin{bmatrix} s^C \\ 1 \end{bmatrix} = s^F. \tag{4.6}$$

This definition of $\widetilde{P}$ was suggested in [3] and denoted by *variant 2*. The definition of Q given in (4.7) apparently fulfills (4.6) and retains the sparsity of P. For the entries Q_{ij}, where i is the index of a fine grid point and j the index of a coarse grid point, we define

$$Q_{ij} := P_{ij} \left(\frac{s_i^F}{\sum\limits_{j \in C_i} P_{ij}} - s_j^C \right), \tag{4.7}$$

with C_i is the set of coarse points in the direct neighborhood of i, i.e., the indices of the columns with nonzero entries in row i of the interpolation P.

Let us finally remark that the GM approach is a hybrid approach in the sense that the coarsening strategy is nodal-based but the interpolation P is unknown based; see Section 4.2 for a description of both methods. The hybrid approach is necessary since we need to have all unknowns in a node to be either coarse or fine in order to perform the interpolation of a smooth error vector. This is obvious since the rotation $[y, -x]$ operates on both unknowns x and y. The unknown based GM interpolation in two dimensions then also writes

$$\widetilde{P} = \begin{bmatrix} P_x & 0 & Q_x \\ 0 & P_y & Q_y \end{bmatrix},$$

and Q_x and Q_y can be computed independently and have the same sparsity as P_x and P_y. In the GM implementation in BoomerAMG an independent truncation of Q and P is possible, to influence the sparsity of the interpolation and thus the operator complexities.

4.4 Numerical Results for irFETI-DP

In this section, we will present numerical results in order to compare the different AMG approaches described before and their performance as preconditioners for the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$. We always use one V-cycle of BoomerAMG in our implementation. The BoomerAMG package, see [68], provides an efficient parallel implementation of the GM approach and the user simply has to provide smooth error vectors of his or her choice on the fine grid. In our case, in two dimensions, we have to provide the rotation s^F of the FETI-DP coarse grid.

In order to provide a fair comparison of the three different methods, we carefully choose an AMG setup where all components have shown the potential to scale up to large scales. We choose the highly scalable HMIS aggressive coarsening scheme, introduced in [30], the long range *extended+i* interpolation method described in [29, 138] and one step of a symmetric SOR/Jacobi smoother. Of course, we use the same setup in all three methods: the unknown approach, the hybrid approach with nodal coarsening, and the GM approach.

If a Dirichlet boundary condition will be applied to the bigger part of the boundary, standard nodal or unknown approaches are known to perform well and we do not expect any additional benefit from the GM approach. This is also confirmed by our numerical results in Section 3.4, where we always used a nodal coarsening. Therefore, we now choose a more difficult problem setup and consider an elasticity problem on a rectangular domain Ω with an aspect ratio of $8:1$, only fixed on one of the smaller sides. A volume force orthogonal to the longer sides is applied. We refer to this problem as *2D beam* problem and a solution considering a linear elastic material is presented in Fig. 4.3. Let us remark that the GM approach already showed a promising performance for such a problem layout in two and three dimensions; see [3] for the results.

We will present parallel results up to $131K$ computational cores for both, the *2D beam* with linear elasticity solved with irFETI-DP and the *2D beam* with nonlinear hyperelasticity solved with irNonlinear-FETI-DP-1. In both cases we choose only primal vertex constrains to build our FETI-DP coarse space.

4.4.1 Linear Elasticity Results

We first consider the linear elasticity problem

$$-2\mu \,\mathrm{div}(\varepsilon(u)) - \lambda \,\mathrm{grad}(\mathrm{div}(u)) = f$$

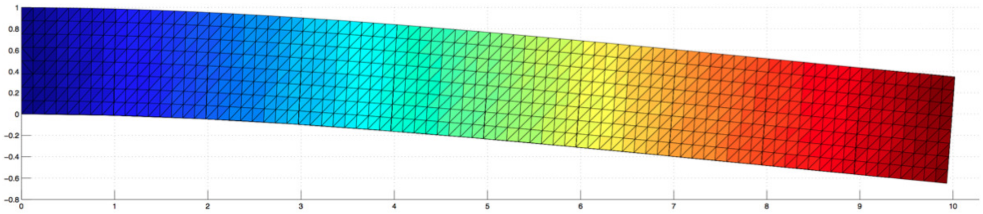

Fig. 4.2: Solution of the *2D beam* considering linear elasticity with $E = 210$ and $\nu = 0.3$.

with material parameters

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \text{ and } \mu = \frac{E}{2(1+\nu)},$$

c.f., [13]. We choose $E = 210$ and $\nu = 0.3$.

Weak scalability results for the ***2D beam model*** problem with linear elasticity are presented in Tab. 4.1 and a graphical representation can be found in Fig. 4.3. These tests are performed on the Vulcan BlueGene/Q and we start with a problem size of 643 602 degrees of freedom on 32 MPI-ranks and scale up to a problem size with 2 621 670 402 degrees of freedom on 131 072 MPI-ranks. Using the total time to solution on 32 cores of the fastest approach (unknown approach) as a baseline, we obtain 62% parallel efficiency for the unknown approach, 76% efficiency for the hybrid approach, and even 95% for the GM approach on 131 072 cores. As expected, we benefit from the exact interpolation of the rotation s^F and obtain both, numerical scalability with respect to the number of Krylov subspace iterations and very good parallel scalability using the GM approach. In fact, the number of Krylov iterations shrinks with the number of subdomains using the GM approach, while it grows using the unknown approach. This effect can be explained by the growing FETI-DP coarse space and thus the increasing benefit from having a good preconditioner $\widehat{S}_{\Pi\Pi}$ or, on the contrary, a deterioration using an insufficient preconditioner. In the problem with 32 subdomains computed on 32 MPI-ranks the coarse problem only consists of 30 degrees of freedom and the effect of the AMG preconditioner on the convergence of irFETI-DP is negligible. Therefore, all three approaches behave similar. Using 131 072 cores and a coarse problem including more than 260 000 degrees of freedom, the GM approach can reduce the time spent in the Krylov method by a factor of 3.5 compared to the unknown approach, and also the time spent in the AMG setup is still small.

					Time	Time
# MPI		$\widetilde{S}_{\Pi\Pi}$		**Time**	**to**	**AMG**
ranks	**d.o.f.**	**preconditioner**	**It.(Cop)**	**GMRES**	**Solution**	**Setup**
32	643 602	U-AMG	76 (1.3)	6.6s	17.6s	0.03s
		H-AMG	85 (1.1)	7.3s	18.3s	0.03s
		H-AMG-GM	87 (1.1)	7.5s	18.5s	0.03s
128	2 567 202	U-AMG	83 (1.5)	7.9s	18.9s	0.05s
		H-AMG	64 (1.3)	6.1s	17.1s	0.04s
		H-AMG-GM	58 (1.3)	5.4s	16.4s	0.04s
512	10 254 402	U-AMG	87 (1.9)	9.0s	20.1s	0.08s
		H-AMG	82 (1.5)	8.3s	19.4s	0.06s
		H-AMG-GM	52 (1.6)	5.3s	16.4s	0.07s
2 048	40 988 802	U-AMG	112 (2.5)	10.9s	22.1s	0.12s
		H-AMG	113 (1.8)	10.7s	21.9s	0.10s
		H-AMG-GM	47 (2.1)	4.5s	15.6s	0.12s
8 192	163 897 602	U-AMG	111 (2.7)	11.1s	22.6s	0.17s
		H-AMG	71 (1.6)	6.8s	18.2s	0.11s
		H-AMG-GM	43 (1.8)	4.1s	15.5s	0.15s
32 768	655 475 202	U-AMG	119 (2.8)	13.0s	25.1s	0.28s
		H-AMG	80 (1.6)	7.9s	19.9s	0.15s
		H-AMG-GM	40 (1.8)	4.0s	16.1s	0.20s
131 072	2 621 670 402	U-AMG	123 (2.9)	14.0s	28.6s	0.46s
		H-AMG	85 (1.6)	8.6s	23.1s	0.25s
		H-AMG-GM	39 (1.8)	4.0s	18.5s	0.33s

Table 4.1: Results and timings for linear irFETI-DP applied to the model problem *2D beam* with one V-cycle of BoomerAMG as preconditioner for the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$. We choose a linear elastic model with material parameters $E = 210$ and $\nu = 0.3$ and discretize the computational domain with quadratic, triangular finite elements. We assign the FETI-DP coarse problem to 5% of the MPI ranks. Here, **U-AMG** stands for the unknown approach, **H-AMG** for the hybrid approach with nodal coarsening using the row-sum norm, and **H-AMG-GM** for the GM approach also using the row-sum norm. In all methods HMIS coarsening, extended+i interpolation, and a symmetric SOR/Jacobi smoother is used. The remaining columns are: **It.(Cop)** the number of GMRES iterations and the AMG operator complexity; **Time GMRES** the runtime spent in GMRES; **Time AMG Setup** the runtime spent in the AMG setup.

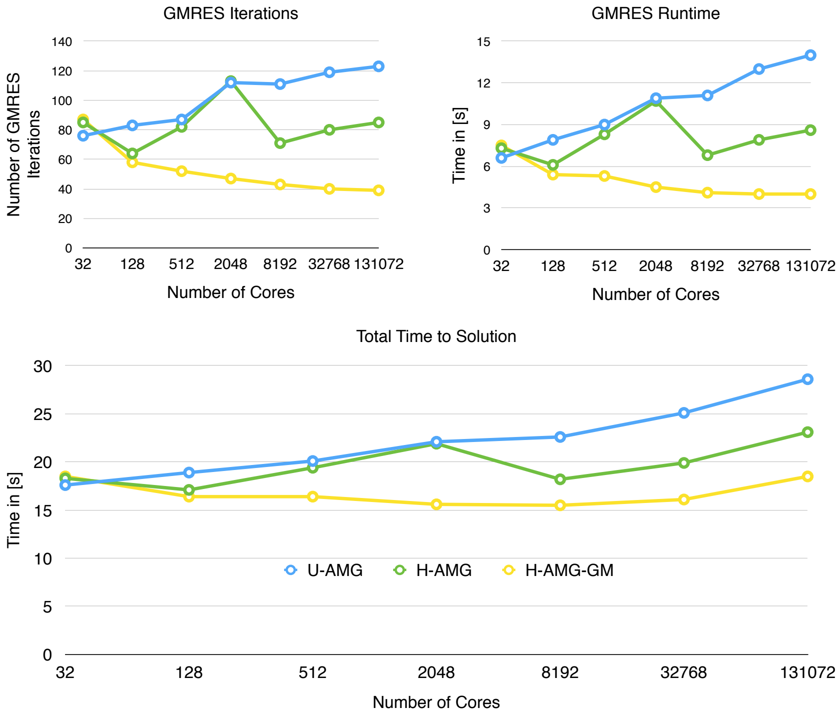

Fig. 4.3: Graphical representation of the results presented in Tab. 4.1 for irFETI-DP applied to the linear elastic beam with different AMG approaches as preconditioner for the FETI-DP coarse problem. The unknown approach is represented by the blue lines (U-AMG), the hybrid approach by the green line (H-AMG), and the GM approach by the yellow line. The figure at the top left shows the number of GMRES iterations. The figure at the top right the GMRES runtime and the figure at the bottom shows the overall runtime.

4.4.2 Nonlinear Hyperelasticity Results

Let us now present weak scalability results for the *2D beam* choosing the nonlinear Neo-Hooke material model introduced in Section 3.4.1. We choose material parameters $E = 210000$ and $\nu = 0.3$ and apply a volume force of 0.01. This nonlinear problem is solved using our inexact reduced Nonlinear-FETI-DP-1 method while applying the different AMG methods in both nonlinear FETI-DP phases, namely the computation of an initial value (Phase 1) and the solution phase (Phase 2). Results for this setup are presented in Table 4.2 and have been performed on the JUQUEEN BlueGene/Q. A graphical representation can be found in Fig. 4.4. As in the linear case, the GM approach shows the potential to save GMRES iterations in Phase 2 and therefore reduces the iteration time as well as the runtime. The effect is not as impressive as in the linear case, since Phase 1 only profits marginally from the optimized preconditioner for the coarse space. This is the case because the iteration time in Phase 1 is negligible. In contrast to linear elasticity, the AMG setup time slightly increases with the number of MPI-ranks and FETI-DP subdomains, especially for the unknown approach. We made several experiments with different AMG setups in order to reduce the time spent in BoomerAMG setup, i.e., using a more drastic truncation of the interpolation P. In fact, this leads to a faster AMG setup but also results in higher GMRES iteration counts. This also shows the superiority of the GM approach in this set of experiments since we have an acceptable AMG setup time as well as fast iterations.

Nevertheless, the imperfect scalability of the AMG setup phase causes a less optimal scalability compared to the linear case. Choosing the time to solution of the fastest inexact method on 32 cores as a baseline ($38.0s$; nodal coarsening), we obtain a parallel efficiency of only 53% for the unknown approach, 64% for the nodal coarsening, and respectable 73% for the GM approach on 131 072 cores.

To obtain an estimation of the general quality of BoomerAMG using the GM approach as a preconditioner for $\widetilde{S}_{\Pi\Pi}$, we also provide a comparison with the exact Nonlinear-FETI-DP-1 approach. The results can also be found in Table 4.2. Let us recall from Section 3.1 that we can expect a similar asymptotical convergence behavior of both methods (exact and inexact reduced FETI-DP) in the case of a good preconditioner $\widehat{S}_{\Pi\Pi}$. The results presented in Table 4.2 are satisfying, since the number of GMRES iterations using irNonlinear-FETI-DP-1 with the GM approach (82) is only slightly higher than the number of iterations using exact Nonlinear-FETI-DP-1 (62) for the problem decomposed into 32 768 subdomains. The poor scalability of exact Nonlinear-FETI-DP-1 is expected, since we use the sequential sparse direct solver UMFPACK [27] to factorize the FETI-DP coarse problem. Choosing a MPI parallel direct solver as, e.g., MUMPS [28], or a thread-parallel solver, as, e.g., PARDISO [97,125,126], might improve the scalability and also the memory efficiency.

In exact FETI-DP methods we have a polylogarithmic condition number bound with respect to H/h; see also (2.11). The distribution of the eigenvalues of the preconditioned system plays a major role for the convergence behavior of iterative methods as CG or GMRES; see Sections 1.1.2 and 1.1.3. We thus would expect logarithmic increasing iteration counts with respect to H/h, at least using exact Nonlinear-FETI-DP-1 and, considering the presented results, also using irNonlinear-FETI-DP-1 with GM approach. Therefore, we present numerical results for the *2D beam* with Neo-Hooke material and increasing H/h in Fig. 4.5. We use 32 768 subdomains and MPI ranks. We start with $H/h = 5$ and increase up to $H/h = 120$. The case with $H/h = 50$ is the same problem as presented in Table 4.2. In Fig. 4.5, we see depicted results for exact Nonlinear-FETI-DP-1 (blue), irNonlinear-FETI-DP-1 with classical AMG (yellow), and irNonlinear-FETI-DP-1 with GM approach (green). The iteration counts at the dots are the measured values. We fit a logarithmic trendline and in the exact case as well as the GM case the results indicate the expected logarithmic behavior. Again, AMG with GM interpolations seems to be a good preconditioner for the FETI-DP coarse problem. Classical AMG with an unknown coarsening approach hardly shows a logarithmic behavior and the measured values do not coincide with the best fitted logarithmic curve.

Let us summarize that the GM approach is a valid method to optimize the performance of inexact reduced FETI-DP methods and leads to much better parallel and numerical scalability results compared to classical AMG methods. This holds for linear elasticity as well as for nonlinear hyperelasticity problems. Nevertheless, in the nonlinear case, the BoomerAMG setup does not scale optimally and it might be necessary to reconsider the choice of several parameters, e.g., the truncation of P and Q, the coarsening method, and the interpolation strategy.

# MPI ranks	d.o.f.	$\widetilde{S}_{\Pi\Pi}$ preconditioner	It.(Cop)	Time GMRES	Time to Solution	Time AMG Setup
32	643 602	U-AMG	161	13.7s	38.9s	0.03s
		H-AMG	152	12.9s	38.0s	0.02s
		H-AMG-GM	174	14.7s	39.8s	0.02s
		exact	57	4.7s	29.5s	-
128	2 567 202	U-AMG	168	15.6s	41.0s	0.03s
		H-AMG	159	14.6s	39.9s	0.03s
		H-AMG-GM	127	11.7s	37.0s	0.03s
		exact	60	5.0s	30.0s	-
512	10 254 402	U-AMG	169	17.3s	43.0s	0.05s
		H-AMG	168	16.7s	42.3s	0.06s
		H-AMG-GM	102	10.3s	35.8s	0.05s
		exact	62	5.3s	30.5s	-
2 048	40 988 802	U-AMG	165	17.6s	44.0s	0.17s
		H-AMG	162	16.6s	42.4s	0.11s
		H-AMG-GM	95	9.8s	35.6	0.11s
		exact	62	5.6s	31.7s	-
8 192	163 897 602	U-AMG	163	19.5s	48.3s	0.61s
		H-AMG	165	17.4s	44.2s	0.25s
		H-AMG-GM	89	9.4s	36.2s	0.25s
		exact	62	7.1s	37.9s	-
32 768	655 475 202	U-AMG	176	23.7s	51.0s	2.23s
		H-AMG	158	17.2s	47.1s	0.71s
		H-AMG-GM	82	9.0s	39.1s	0.82s
		exact	62	14.1s	67.1s	-
131 072	2 621 670 402	U-AMG	166	24.5s	72.3s	6.1s
		H-AMG	147	16.4s	59.1s	2.6s
		H-AMG-GM	81	9.1s	52.6s	2.4s
		exact	OoM	OoM	OoM	-

Table 4.2: Results and timings for inexact reduced Nonlinear-FETI-DP-1 applied to the *2D beam* model problem with one V-cycle of BoomerAMG as preconditioner for the FETI-DP coarse problem $\widetilde{S}_{\Pi\Pi}$. We choose a Neo-Hooke model with material parameters $E = 210000$ and $\nu = 0.3$. The remaining notation and the AMG setup is chosen as in Table 4.1. We additionally provide results using **exact** Nonlinear-FETI-DP-1.

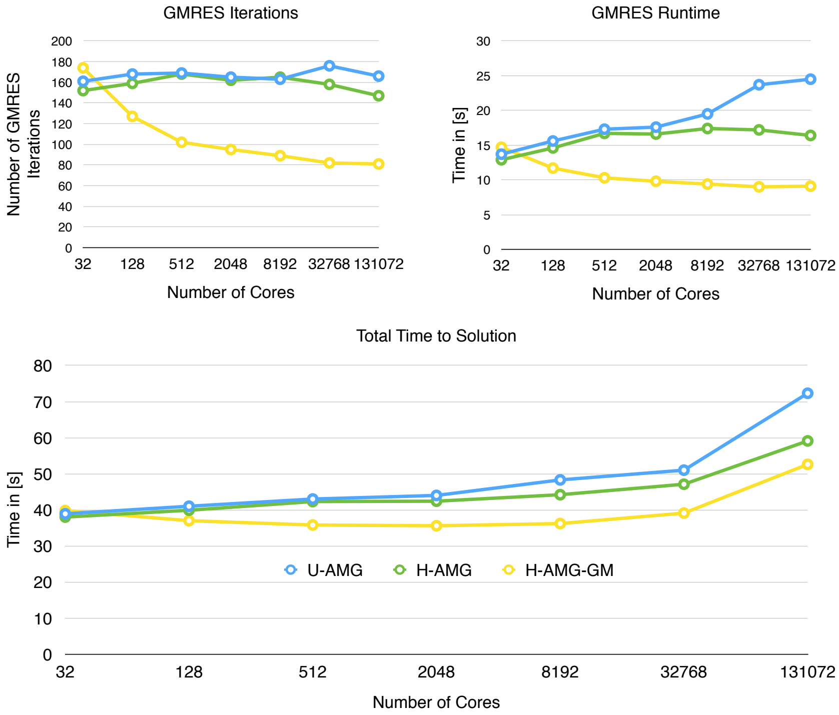

Fig. 4.4: Graphical representation of the results presented in Table 4.2 for irNonlinear-FETI-DP-1 applied to the nonlinear hyperelastic beam with different AMG approaches as preconditioner for the FETI-DP coarse problem. The unknown approach is represented by the blue lines (U-AMG), the hybrid approach by the green line (H-AMG), and the GM approach by the yellow line. The figure at the top left shows the number of GMRES iterations, the figure at the top right the GMRES runtime, and the figure at the bottom shows the overall runtime.

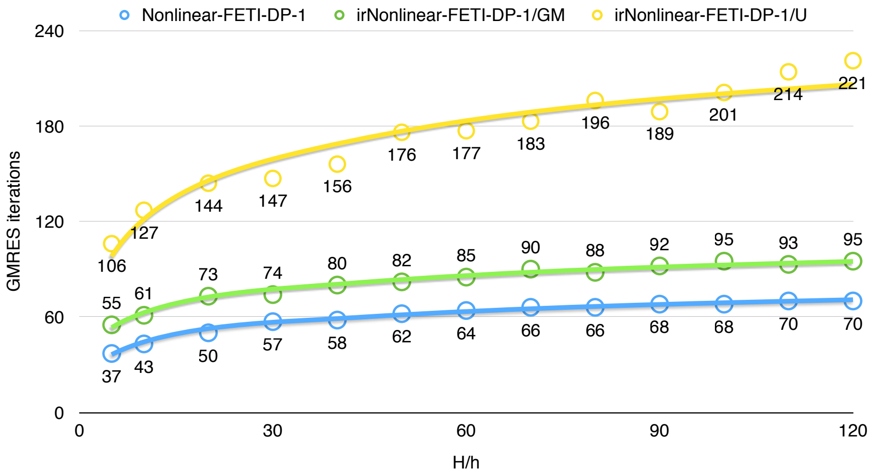

Fig. 4.5: Number of total GMRES iterations with respect to H/h. The nonlinear *2D beam* is decomposed into 32 768 subdomains and the computations are performed using 32 768 JUQUEEN cores. The remaining parameters are chosen as in Table 4.2. The dots are the measured values and the line is the best possible logarithmic fit to the measurements.

4.5 Inexact FETI-DP

If the chosen AMG method is a strong preconditioner for the FETI-DP coarse problem, we may be able to eliminate all direct solvers and apply AMG to the complete system $D\widetilde{K}$ instead of first reducing $D\widetilde{K}$ to the primal Schur complement matrix $\widetilde{S}_{\Pi\Pi}$. This approach is denoted by iFETI-DP (inexact FETI-DP), first described in [84]. We will now provide a brief description of a nonlinear version of iFETI-DP and present results for the ***2D beam*** problem with linear elastic material.

4.5.1 Inexact Nonlinear FETI-DP

Let us recall the linearization of the nonlinear FETI-DP master system from (2.36)

$$\begin{bmatrix} D\widetilde{K}(\tilde{u}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u} \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} \widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f} \\ B\tilde{u} \end{bmatrix}. \tag{4.8}$$

Here, for simplicity, we drop the index k of the current Newton iterate. Following the notation in [84] we define

$$\mathscr{A} := \begin{bmatrix} D\widetilde{K}(\tilde{u}) & B^T \\ B & 0 \end{bmatrix},$$

$$\mathscr{F} := \begin{bmatrix} \widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f} \\ B\tilde{u} \end{bmatrix},$$

and

$$x := \begin{bmatrix} \delta\tilde{u} \\ \delta\lambda \end{bmatrix}.$$

Instead of eliminating any variables, we can directly apply a Krylov subspace method, e.g., GMRES, to the preconditioned system

$$\mathscr{B}_L^{-1}\mathscr{A}x = \mathscr{B}_L^{-1}\mathscr{F} \tag{4.9}$$

and thus solve iteratively. The block triangular preconditioner $\mathscr{B}_L$ is defined by

$$\mathscr{B}_L := \begin{bmatrix} \widehat{K}^{-1} & 0 \\ M^{-1}B\widehat{K}^{-1} & M^{-1} \end{bmatrix},$$

where $\widehat{K}^{-1}$ is a good preconditioner for $D\widetilde{K}(\tilde{u})$ and M^{-1} is again one of the standard FETI-DP preconditioners. As in inexact reduced FETI-DP, the application of $\widehat{K}^{-1}$ will consist of few cycles of a parallel AMG method, but this time applied to the full system $D\widetilde{K}(\tilde{u})$. Let us remark that we no longer have the inverse $DK(\tilde{u})_{BB}^{-1}$ in our system matrix and we thus do not need any sparse direct solvers in an application of $\mathscr{A}$. That should reduce the amount of required memory. In principal, we moved an approximation of the inverse $DK(\tilde{u})_{BB}^{-1}$ from the system matrix to the preconditioner since we apply an AMG method to the block system

$$D\widetilde{K}(\tilde{u}) := \begin{bmatrix} DK_{BB}(\tilde{u}) & D\widetilde{K}_{B\Pi}(\tilde{u}) \\ D\widetilde{K}_{\Pi B}(\tilde{u}) & D\widetilde{K}_{\Pi\Pi}(\tilde{u}) \end{bmatrix}.$$

Therefore, the numerical performance of the AMG method is even more important than in the inexact reduced (nonlinear) FETI-DP variants.

Nevertheless, if we use the common Dirichlet preconditioner $M^{-1} := M_{FETI_D}^{-1}$, we will need a sparse direct solver for the inverse operators $\left(DK_{II}^{(i)}(\tilde{u})\right)^{-1}$ on the interior part of the subdomain Ω_i, $i = 1,...,N$; compare also (2.9) for the definition of $M_{FETI_D}^{-1}$. To obtain a purely iterative method without LU factorizations, we can replace an application of $\left(DK_{II}^{(i)}(\tilde{u})\right)^{-1}$ by some cycles of a local AMG method. We denote this modified Dirichlet preconditioner by $M_{FETI_{D/AMG}}^{-1}$. Let us note that this approach does not guarantee to be spectral equivalent to the (exact) Dirichlet preconditioner unless the interior system is solved accurately enough. Nevertheless, this modified preconditioner often leads to appropriate results; see also [81]. Let us remark that also the usage of the lumped preconditioner $M_{FETI_L}^{-1}$ is possible, but this approach does not appear to be robust. See (2.10) for a definition of $M_{FETI_L}^{-1}$.

4.5.2 Algorithmic Description and Implementation Details

We implemented the inexact nonlinear FETI-DP method in PETSc 3.4.3 using C/C++ and MPI. The basic infrastructure is more or less the same as in our inexact reduced FETI-DP implementation introduced in Chapter 3. The major difference is the need for an assembly of matrix $D\widetilde{K}(\tilde{u})$, which supersedes LU factorizations of $DK_{BB}(\tilde{u})$ and the assembly of the primal Schur complement $\widetilde{S}_{\Pi\Pi}$; see Sections 3.3.1 and 3.3.2 for a description of the building blocks. We decided to implement the matrix $D\widetilde{K}(\tilde{u})$ and the jump operator B as MPI parallel sparse matrices of the type *MPIAIJ*, which is provided by PETSc. All rows of $D\widetilde{K}(\tilde{u})$ corresponding to the interior and interface nodes of the same subdomain Ω_i are distributed to the same MPI rank, i.e., the local subdomain block

$$\begin{bmatrix} DK_{BB}^{(i)}(\tilde{u}) & D\widetilde{K}_{B\Pi}^{(i)}(\tilde{u}) \end{bmatrix}$$

is assigned to one MPI rank. The rows corresponding to the globally assembled FETI-DP coarse space are equally distributed to all ranks and thus we do not obtain the typical block structure

$$D\widetilde{K}(\tilde{u}) := \begin{bmatrix} DK_{BB}(\tilde{u}) & D\widetilde{K}_{\Pi B}^{T}(\tilde{u}) \\ D\widetilde{K}_{\Pi B}(\tilde{u}) & D\widetilde{K}_{\Pi\Pi}(\tilde{u}) \end{bmatrix}$$

in our implementation. We always try to distribute a primal variable to one of the MPI ranks handling a neighboring subdomain. This strategy should reduce communication. The rows of B^T are distributed equivalently.

As preconditioner for $D\widetilde{K}(\tilde{u})$ we always use one V-cycle of BoomerAMG with GM interpolation. Of course, we have to provide the rigid body modes on the finite element space $\widetilde{W}$, i.e., the rotation of the coarse space and the subdomain nodes. We also present an algorithmic description of inexact nonlinear FETI-DP in form of a pseudocode in Fig. 4.6.

4.5.3 Numerical Results

Once more, we consider the linear elastic *2D beam* and perform weak scalability experiments for the inexact FETI-DP approach. We consider the same parameters and problem details as described in Section 4.4.1 and Table 4.1. We always use AMG with GM interpolations since classical AMG approaches appear to be not robust and strong enough as preconditioner for $D\widetilde{K}(\tilde{u})$.

In Table 4.3, we compare iFETI-DP with the standard Dirichlet preconditioner $M_{FETI_D}^{-1}$, where the application of $DK_{II}^{-1}(\tilde{u})$ is performed by a forward backward solve in a sparse direct solver package, to the modified Dirichlet preconditioner $M_{FETI_{D/AMG}}^{-1}$, where we replace the LU factorization of $DK_{II}(\tilde{u})$ by one BoomerAMG V-cycle. The computations have been performed on the Vulcan BlueGene/Q. The numerical scalability with respect to the number of GMRES iterations is convincing and the iteration count for both methods is in the range of 23 and 35. In fact, the number of iterations is slightly decreasing when scaling up the number of subdomains. The replacement of the LU factorization by an AMG V-cycle does thus not deteriorate the preconditioner in this experimental setup. In principle, the scalability of both methods is comparable, but choosing the modified Dirichlet preconditioner is always slightly faster since the AMG setup beats the LU factorization; see Table 4.4. Overall, we have a parallel

```
Inexact Nonlinear FETI-DP Method

Init: $\tilde{u}^{(0)} \in \widetilde{W}$
for $k = 0, ..., convergence$
    build: $\widetilde{K}(\tilde{u}^{(k)})$, $D\widetilde{K}(\tilde{u}^{(k)})$, and $M^{-1}$
    iterative Krylov solve for $x = [\delta\tilde{u}^{(k)T}, \delta\lambda^{(k)}]$ using left
    preconditioner $\mathscr{B}_L^{-1} := \mathscr{B}_L^{-1}(\widehat{K}^{-1}, M^{-1})$:
        $\mathscr{A}x = \mathscr{F}$        // (see eq. (4.9))

    compute: steplength $\alpha^{(k)}$
    update:
    $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\,\delta\tilde{u}^{(k)}$
    $\lambda^{(k+1)} := \lambda^{(k)} - \alpha^{(k)}\,\delta\lambda^{(k)}$
end
```

Fig. 4.6: Pseudocode of the **inexact nonlinear FETI-DP algorithm**. The application of $\widehat{K}^{-1}$ will consist of cycles of a parallel AMG method.

efficiency of 73% on 131 072 cores using the classical Dirichlet preconditioner and 82% using the modified preconditioner. As a baseline, we choose the time to solution of 21.0 seconds of the fastest approach on 32 MPI ranks, which is iFETI with the modified Dirichlet preconditioner.

In general, the weak parallel scalability is convincing, but we see a rapid deterioration when scaling from 32 786 to 131 072 MPI ranks. To investigate this effect we provide detailed time measurements in Table 4.4. The time to solution splits up in five major parts. The first part is the finite element assembly of the local tangential matrices $DK^{(i)}(\tilde{u})$ and perfectly scalable. This is expected and has already been observed in the inexact reduced FETI-DP methods; see, e.g., Section 3.4.4. For the second part, namely the assembly of the global and MPI parallel matrix $D\widetilde{K}(\tilde{u})$, MPI communication is necessary and starts to be a bottleneck on 131 072 cores. The time spent in this assembly step takes approximately $1.75s$ on 131 072 cores instead of $0.33s$ on 32 cores. This is one reason for the increased runtime to solution on 131 072 cores. We observed the same effect in the assembly of the primal Schur complement $\widetilde{S}_{\Pi\Pi}$ in irNonlinear-FETI-DP-1;

see, e.g., Fig. 3.9. A mentionable amount of time is spent in the BoomerAMG setup for the matrix $D\widetilde{K}(\tilde{u})$. This is the third part and denoted by ***Time Setup*** $\widehat{K}^{-1}$ in Table 4.4. Here, we also see a slight increase in the runtime on 131 072 cores, which also affects the overall scalability. Also the time spent in the setup of the Dirichlet or modified Dirichlet preconditioner M^{-1} is slightly increased. In Table 4.4, it can be seen that the BoomerAMG setup used in the setup of $M^{-1}_{FETI_{D/AMG}}$ is always faster than the computation of the LU factorizations in the setup of $M^{-1}_{FETI_D}$. On the other hand, the forward/backward solves in an application of $M^{-1}_{FETI_D}$ are slightly cheaper than the application of BoomerAMG. This causes a slightly faster fifth part of the runtime, namely a slightly faster GMRES iteration.

All these observations are also graphically presented; see Figure 4.7 for iFETI with the Dirichlet preconditioner and Figure 4.8 for iFETI with the modified Dirichlet preconditioner.

To summarize, we have seen a highly scalable inexact nonlinear FETI-DP implementation using a modified Dirichlet preconditioner on the interface, where all sparse direct solvers have been eliminated in order to reduce the memory consumption. Combined with the GM interpolation approach iFETI-DP appears to be robust for linear elasticity problems and may be an alternative to inexact reduced FETI–DP variants.

# MPI ranks	**d.o.f.**	M^{-1}	**It.**	**Time to Solution**	**Eff.**
32	643 602	$M^{-1}_{FETI_D}$	34	23.73s	88%
		$M^{-1}_{FETI_{D/AMG}}$	32	21.00s	100%
128	2 567 202	$M^{-1}_{FETI_D}$	35	24.57s	85%
		$M^{-1}_{FETI_{D/AMG}}$	32	21.12s	99%
512	10 254 402	$M^{-1}_{FETI_D}$	31	23.91s	88%
		$M^{-1}_{FETI_{D/AMG}}$	30	20.79s	101%
2 048	40 988 802	$M^{-1}_{FETI_D}$	27	22.96s	91%
		$M^{-1}_{FETI_{D/AMG}}$	26	19.70s	107%
8 192	163 897 602	$M^{-1}_{FETI_D}$	27	23.40s	90%
		$M^{-1}_{FETI_{D/AMG}}$	27	20.41s	103%
32 786	655 475 202	$M^{-1}_{FETI_D}$	24	23.82s	88%
		$M^{-1}_{FETI_{D/AMG}}$	23	20.34s	103%
131 072	2 621 670 402	$M^{-1}_{FETI_D}$	24	28.66s	73%
		$M^{-1}_{FETI_{D/AMG}}$	25	25.79s	82%

Table 4.3: Same model problem as in Table 4.1; solved with **iFETI-DP**; one V-cycle of BoomerAMG with GM interpolation is used as preconditioner $\widehat{K}^{-1}$; **It.** denotes the number of GMRES iterations; the baseline of the parallel efficiency **Eff.** is the fastest time to solution on 32 MPI ranks ($21.00s$).

		Time	**Time**	**Time**	**Time**	**Time**
# MPI		**Assembly**	**Assembly**	**Setup**	**Setup**	**GMRES**
ranks	M^{-1}	$DK^{(i)}$	$D\widetilde{K}$	$\widehat{K}^{-1}$	M^{-1}	
32	$M^{-1}_{FETI_D}$	6.15s	0.33s	3.38s	4.66s	8.69s
	$M^{-1}_{FETI_{D/AMG}}$	6.15s	0.33s	3.36s	1.17s	9.44s
128	$M^{-1}_{FETI_D}$	6.13s	0.33s	3.43s	5.07s	9.05s
	$M^{-1}_{FETI_{D/AMG}}$	6.14s	0.34s	3.42s	1.17s	9.50s
512	$M^{-1}_{FETI_D}$	6.16s	0.33s	3.47s	5.25s	8.13s
	$M^{-1}_{FETI_{D/AMG}}$	6.19s	0.35s	3.48s	1.18s	9.04s
2 048	$M^{-1}_{FETI_D}$	6.16s	0.35s	3.54s	5.25s	7.08s
	$M^{-1}_{FETI_{D/AMG}}$	6.18s	0.35s	3.53s	1.21s	7.86s
8 192	$M^{-1}_{FETI_D}$	6.23s	0.39s	3.63s	5.37s	7.13s
	$M^{-1}_{FETI_{D/AMG}}$	6.25s	0.39s	3.64s	1.26s	8.22s
32 786	$M^{-1}_{FETI_D}$	6.25s	0.62s	3.85s	5.63s	6.43s
	$M^{-1}_{FETI_{D/AMG}}$	6.23s	0.62s	3.85s	1.50s	7.08s
131 072	$M^{-1}_{FETI_D}$	6.26s	1.74s	4.63s	6.64s	6.62s
	$M^{-1}_{FETI_{D/AMG}}$	6.27s	1.79s	4.63s	2.48s	7.87

Table 4.4: Same model problem as in Table 4.1; solved with **iFETI-DP**; one V-cycle of BoomerAMG with GM interpolation is used as preconditioner $\widehat{K}^{-1}$; detailed timings for the assembly of the local operator $DK^{(i)}$, for the assembly of the MPI parallel matrix $D\widetilde{K}$, the BoomerAMG setup phase (Setup $\widehat{K}^{-1}$), the setup of the preconditioner M^{-1}, and the GMRES iteration.

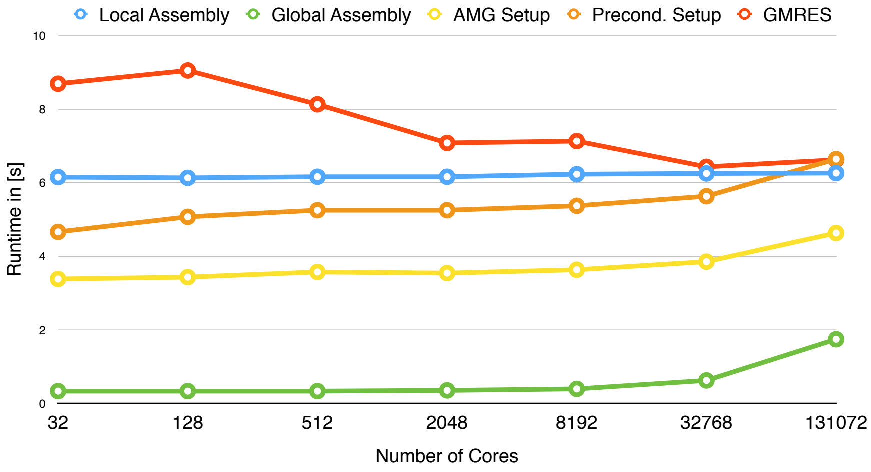

Fig. 4.7: Representation of the detailed timings for **iFETI-DP** with preconditioner $M^{-1}_{FETI_D}$ presented in Table 4.4; **Local Assembly** is the assembly time of $DK^{(i)}(\tilde{u})$; **Global Assembly** is the assembly time of $D\widetilde{K}(\tilde{u})$; **AMG Setup** is the BoomerAMG setup time for $\widehat{K}^{-1}$; **Precond. Setup** is the setup time for the preconditioner $M^{-1}_{FETI_D}$; **GMRES** is the iteration time.

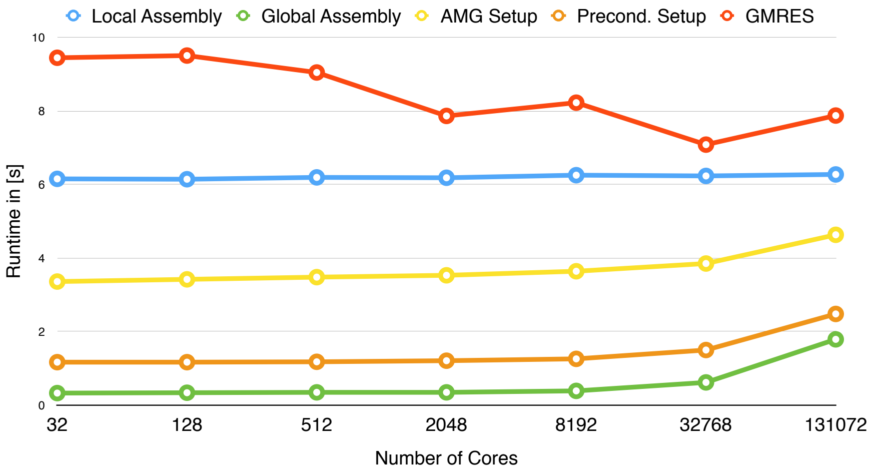

Fig. 4.8: Representation of the detailed timings for **iFETI-DP** with preconditioner $M^{-1}_{FETI_{D/AMG}}$ presented in Table 4.4; **Local Assembly** is the assembly time of $DK^{(i)}$; **Global Assembly** is the assembly time of $D\widetilde{K}$; **AMG Setup** is the BoomerAMG setup time for $\widehat{K}^{-1}$; **Precond. Setup** is the setup time for the preconditioner $M^{-1}_{FETI_{D/AMG}}$; **GMRES** is the iteration time.

5 A Parallel Implementation of the FE^2 Method

The FE^2 method, see, e.g., [49,96,110,127–129], is a computational micro-macro scale bridging approach directly incorporating micromechanics in macroscopic simulations. In this approach, a microscopic boundary value problem based on the definition of a representative volume element (RVE) is solved at each macroscopic Gauß integration point. Then, volumetric averages of microscopic stress distributions are returned to the macroscopic level, which replaces a phenomenological material law at the macro scale.

As already announced in the introduction we use an inexact reduced FETI-DP approach to solve the microscopic problems. Thus, we obtain three levels of parallelism. On the first level, we have independent RVEs at each Gauß integration point of the macroscopic problem. Then, on the second level we provide a FETI-DP type domain decomposition of each RVE, and on a third level we use BoomerAMG to parallelize the solution of the FETI-DP coarse problem. We refer to these types of methods as FE^{2}TI methods. In this chapter we provide a brief description of the FE^{2}TI method and our implementation. We present parallel results scaling up to the full JUQUEEN BlueGene/Q machine at Forschungszentrum Jülich which qualified the FE^{2}TI code for the HighQ club; see [89]. The computational results have been obtained during the ***JUQUEEN Extreme Scaling Workshop 2015*** and are partially published in [79].

5.1 Description of the Method

We will first provide a brief description of the FE^2 scale bridging approach; see [49, 128] for a detailed description. We will follow the notation in [128] and briefly describe the derivation of a consistent tangent modulus; cf. [128].

Let us first denote the characteristic length of a typical deformation problem on the macroscopic scale with L and the characteristic length on the microscale with l. We assume, that the microscopic and heterogeneous structure of the material can be resolved in the scale l and that

$$L \gg l.$$

Additionally, we assume that we have a representative volume element (RVE), which can effectively describe the microscopic and heterogeneous material properties on the macroscale.
We discretize the given macroscopic boundary value problem on the domain $\overline{\mathscr{B}}$ in the scale of L without considering any microscopic structure. Then, at each Gauß point of the macroscopic finite elements a microscopic boundary value problem is discretized in the scale of the microstructure l, using the definition of the RVE. Let us remark that the boundary conditions on the microscale are induced from the macroscopic deformation gradient at the corresponding Gauß point. Throughout this chapter, we will mark macroscopic quantities with bars, as, e.g., the deformation gradient with $\overline{F}$ and the first Piola-Kirchhoff stress tensor with $\overline{P}$. On the microscale we simply use P for the stress and F for the deformation gradient. On the macroscale we do not consider any phenomenological law and replace it by volumetric averages of quantities on the microscale. In Fig. 5.1 one can find a schematic illustration of the homogenization step. Exemplary results of FE^2 calculations can be found in Fig. 5.2.

Let us introduce the microscopic boundary value problem in a reference configuration $\mathscr{B}_0$ with corresponding reference variables X. For a deformed configuration $\mathscr{B}$, the deformation of our reference configuration writes $\varphi : \mathscr{B}_0 \to \mathscr{B}$ and the deformation gradient is then defined by $F := \nabla\varphi$. The balance of momentum in a weak formulation with a variational function δx then writes

$$-\int_{\mathscr{B}_0} \delta x \cdot (\mathrm{Div}_X\, P(F))\, dV = 0, \tag{5.1}$$

with the first Piola-Kirchhoff stress tensor $P(F)$. In contrast to the macroscale, the relation between P and F is described by a phenomenological material law, as, e.g. the Neo-Hooke hyperelasticity model:

$$P(F) = \mu(F + F^{-T}) + \lambda \ln(\det(F))F^{-T}. \tag{5.2}$$

The boundary conditions on the microscale are induced from the macroscopic deformation gradient $\overline{F}$ at the corresponding Gauß integration point. In the case of Dirichlet conditions we simply have $x := \overline{F}X$ for each boundary node $X \in \partial\mathscr{B}_0$. Here, X are the variables in the reference configuration $\mathscr{B}_0$ and x the

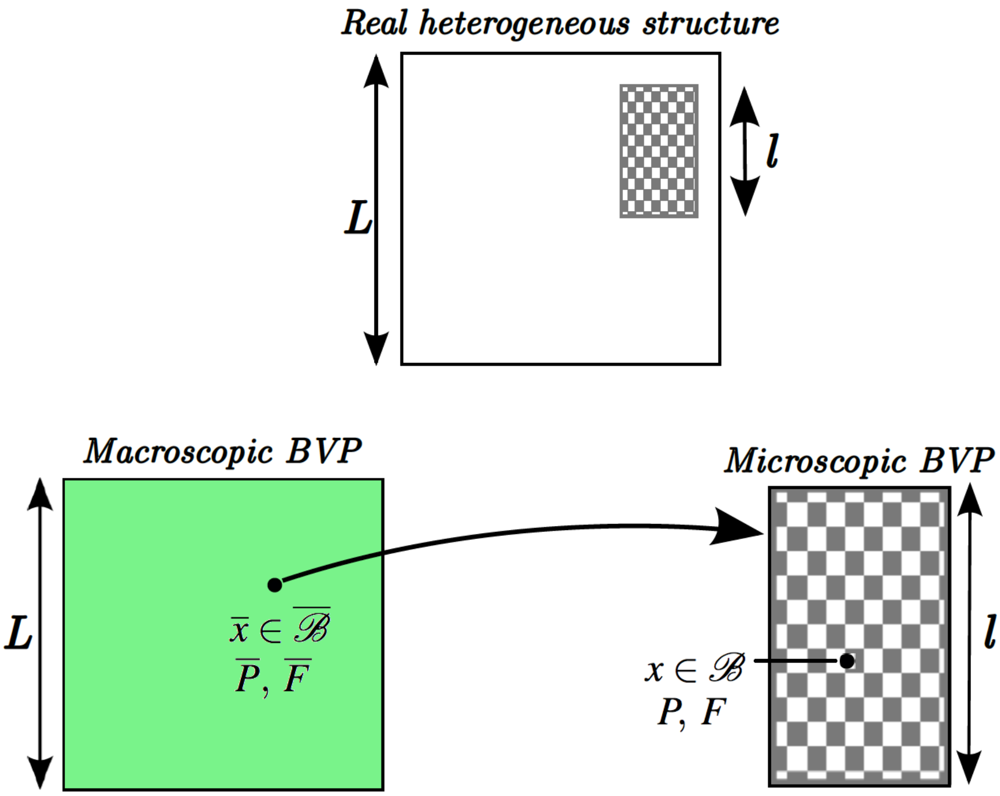
Real heterogeneous structure
L
l
Macroscopic BVP
L
$\overline{x} \in \overline{\mathscr{B}}$
$\overline{P}, \overline{F}$
Microscopic BVP
$x \in \mathscr{B}$
P, F
l

Fig. 5.1: Illustration of the FE^2 homogenization approach. **Top:** Realistic macroscopic boundary value problem of length scale L. A representative volume element describing the microstructure is of length scale l and we have $L \gg l$. **Left:** Macroscopic and simplified boundary value problem on the domain $\overline{\mathscr{B}}$ with quantities $\overline{P}$ and $\overline{F}$. The deformation gradient $\overline{F}$ induces the boundary conditions of the microscopic BVP on the right. **Right:** Microscopic boundary value problem on domain $\mathscr{B}$.

variables in the deformed configuration $\mathscr{B}$.

We can analogously formulate the macroscopic problem in a given reference configuration $\overline{\mathscr{B}}_0$ and reference variables $\overline{X}$. Again, the balance of momentum in the weak formulation with a test function $\delta\overline{x}$ writes

$$-\int_{\overline{\mathscr{B}}_0} \delta\overline{x} \cdot \left(\mathrm{Div}_{\overline{X}}\, \overline{P}(\overline{F}) + \overline{f}\right) dV = 0, \tag{5.3}$$

with some external load $\overline{f}$.
On the macroscale we do not consider any material law. The first Piola-Kirchhoff stress tensor $\overline{P}$ at a macroscopic Gauß point is obtained as a volumetric average

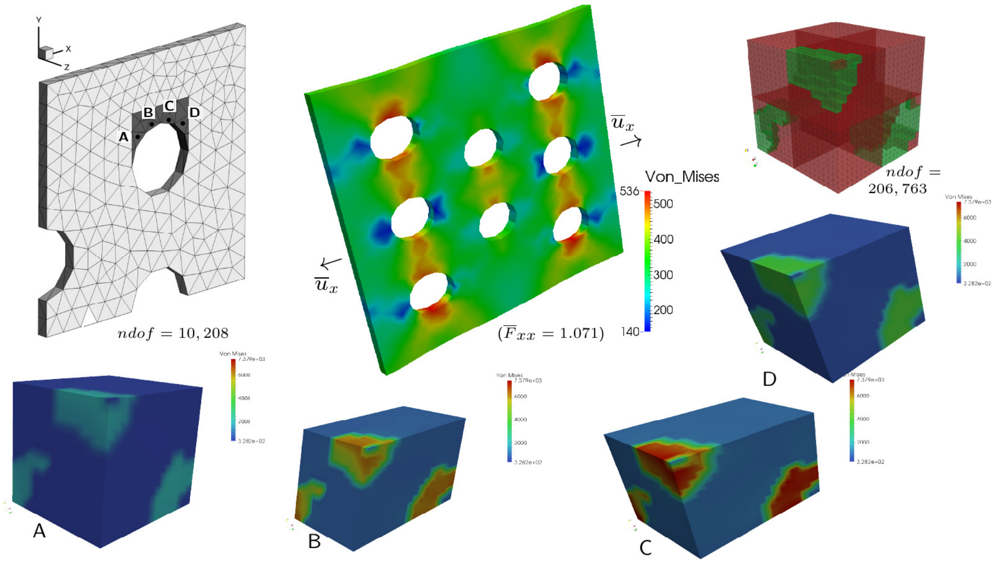

Fig. 5.2: In the FE^2 computational scale bridging method, in each macroscopic Gauß point a microscopic problem is solved; published in [79].

over the Piola-Kirchhoff stresses of the corresponding RVE and thus we have

$$\overline{P} := \frac{1}{V}\int_{\mathscr{B}_0} P(F)\, dV. \tag{5.4}$$

In order to assemble and solve a linearized macroscopic problem, a macroscopic tangent modulus $\overline{\mathbb{A}}$ is required at each Gauß integration point. The tangent modulus is, as usual, simply the derivative of $\overline{P}$ with respect to $\overline{F}$ and thus writes

$$\overline{\mathbb{A}} := \frac{\partial \overline{P}}{\partial \overline{F}} = \frac{\partial}{\partial \overline{F}}\left(\frac{1}{V}\int_{\mathscr{B}_0} P(F) dV\right). \tag{5.5}$$

Let us remark, that $\overline{F}$ is constant on the corresponding RVE. We consider a decomposition of $F =: \overline{F} + \widetilde{F}$ into $\overline{F}$ and a fluctuating part $\widetilde{F}$, which leads, with the help of the chain rule to

$$\overline{\mathbb{A}} := \frac{\partial \overline{P}}{\partial \overline{F}} = \frac{\partial}{\partial \overline{F}}\left(\frac{1}{V}\int_{\mathscr{B}_0} P(F)\, dV\right) \tag{5.6}$$

$$= \frac{1}{V}\int_{\mathscr{B}_0} \frac{\partial P(F)}{\partial F} : \frac{\partial \overline{F} + \widetilde{F}}{\partial \overline{F}}\, dV \tag{5.7}$$

$$= \frac{1}{V}\int_{\mathscr{B}_0} \mathbb{A}\, dV + \frac{1}{V}\int_{\mathscr{B}_0} \mathbb{A} : \frac{\partial \widetilde{F}}{\partial \overline{F}}\, dV. \tag{5.8}$$

The first additive part in (5.8) is simply a volumetric average over the tangent moduli $\mathbb{A}$ of the microscopic problem, while the computation of the second part requires some additional effort. Let us remark that we only have to compute $\overline{\boldsymbol{P}}$ and $\overline{\mathbb{A}}$ after convergence on the microscale is reached. Therefore, we can assume an equilibrium state of the weak formulation in (5.1). Exploiting this fact, Schröder reformulated $\overline{\mathbb{A}}$ in [128] and obtained a discrete and overall consistent tangent modulus

$$\overline{\mathbb{A}}^h := \frac{1}{V}\left(\sum_{T\in\tau}\int_T \mathbb{A}^h \, dV\right) - \frac{1}{V}\boldsymbol{L}^T \left(\boldsymbol{DK}\right)^{-1} \boldsymbol{L}. \tag{5.9}$$

It is left to explain several variables. First, let $\boldsymbol{\tau}$ be the decomposition of $\mathscr{B}_0$ into finite elements and $\mathbb{A}^h$ the discrete microscopic tangent modulus. Then,

$$\frac{1}{V}\left(\sum_{T\in\tau}\int_T \mathbb{A}^h \, dV\right)$$

is simply the discrete representation of

$$\frac{1}{V}\int_{\mathscr{B}_0} \mathbb{A} \, dV$$

in (5.8). The second part in (5.9)

$$\frac{1}{V}\boldsymbol{L}^T \left(\boldsymbol{DK}\right)^{-1} \boldsymbol{L}$$

is the a discrete reformulation of

$$\frac{1}{V}\int_{\mathscr{B}_0} \mathbb{A} : \frac{\partial \widetilde{\boldsymbol{F}}}{\partial \overline{\boldsymbol{F}}} \, dV$$

exploiting the balance of momentum on the microscale; see [128] for the derivation. Here, we have the tangential matrix $\boldsymbol{DK}$ of the microscopic boundary value problem assembled in the usual way from the tangential matrices

$$k_T := \int_T \mathbb{B}_T^T \mathbb{A}^h \mathbb{B}_T \, dV,$$

on the finite elements $\boldsymbol{T} \in \boldsymbol{\tau}$, and $\mathbb{B}_T$ are the derivatives of the shape functions of $\boldsymbol{T}$. The matrix $\boldsymbol{L}$ is assembled analogously from the element matrices

$$l_T := \int_T \mathbb{A}^h \mathbb{B}_T \, dV.$$

Let us remark that $\boldsymbol{L}$ has the dimension $n \times s$, where n is the number of degrees of freedom in the RVE and we have $s = 4$ in two spatial dimensions and $s = 9$ in three spatial dimensions. We finally provide an algorithmic description of the

FE^{2}TI method in Fig. 5.3, where a FETI-DP type method is used to solve the microscopic RVEs.

5.2 Algorithmic Description and Implementation Remarks

Let us summarize the FE^{2}TI approach. We have a macroscopic boundary value problem and at each Gauß integration point one nonlinear RVE has to be solved using a FETI-DP type domain decomposition method. For the parallel solution of the microproblems a subset of computational cores has to be provided at each Gauß point of the macroscopic level. This is realized by splitting the *MPI_COMM_WORLD* communicator into subcommunicators.

In general, the macroscopic problem will be nonlinear since it inherits the nonlinear properties of the phenomenological material law on the microscale. This leads to a macroscopic iteration and in each macroscopic iteration step a full simulation on the microscale has to be performed.

For a realistic simulation of steel deformation, the macroscopic load has to be applied in many small load steps. In the following descriptions and our scalability studies we only consider one macroscopic load step in order to save computational time. This is reasonable, since we are primarily interested in the scalability of the method and our implementation. Of course, we ignore the effect of a load inbalance which might occur in later load steps. This is possible due to the fact that the microscopic problems in different Gauß points may vary in their difficulties, as, e.g., a local plastic behavior.

We provide an algorithmic description of one macroscopic load step in Fig. 5.3.

Let us describe our implementation in some more details. We implemented the FE^{2}TI method in PETSc 3.5.2 using C/C++ and MPI as parallelization paradigm. The macroscopic problem is discretized with piecewise linear triangular elements (P1) in 2D and with piecewise trilinear brick elements (Q1) in 3D. We perform a sequential direct solve using UMFPACK or MUMPS for the macroscopic problem. We solve the macro-problem on all cores redundantly and thus have to provide the macroscopic finite element mesh on all cores. This is reasonable due to the small macroscopic problem sizes; see also Section 5.3. Considering larger problems on even larger machines than the JUQUEEN, a parallel implementation of the macroscopic problem should be considered. A parallel direct solver as well as an AMG method or a domain decomposition method are reasonable variants.

For each Gauß point of the macroscopic problem and thus for each microscopic

Repeat until convergence:

1. Apply boundary conditions to RVE based on macroscopic deformation gradient: Enforce $x = \overline{F}X$ on the boundary of the microscopic problem $\partial\mathscr{B}$ in the case of Dirichlet constraints.
2. Solve microscopic nonlinear boundary value problem using (ir)FETI-DP or related methods.
3. Compute and return macroscopic stresses as volumetric average over microscopic stresses P^h:
$$\overline{P}^h = \frac{1}{V} \sum_{T\in\tau} \int_T P^h dV.$$
4. Compute and return macroscopic tangent moduli as average over microscopic tangent moduli $\mathbb{A}^h$:
$$\overline{\mathbb{A}}^h = \frac{1}{V} \left(\sum_{T\in\tau} \int_T \mathbb{A}^h \, dV \right) - \frac{1}{V} L^T \left(DK\right)^{-1} L$$
5. Assemble tangent matrix and right hand side of the linearized macroscopic boundary value problem using $\overline{P}^h$ and $\overline{\mathbb{A}}^h$.
6. Solve linearized macroscopic boundary value problem.
7. Update macroscopic deformation gradient $\overline{F}$.

Fig. 5.3: Algorithmic description of the FE^{2}TI approach. Overlined letters denote macroscopic quantities.

problem we introduce a separate MPI-Communicator. In our implementation, we use *MPI_Comm_split* to create the subcommunicators. Inter-communicator communication is not necessary during the microscopic solves and the averaging of the different microscopic quantities; see points 2.-4. in Fig. 5.3. To solve the microscopic BVPs we use a Newton-Krylov-irFETI-DP method and the implementation described in Chapter 3. In order to compute $L^T\,(DK)^{-1}L$, see (5.9), we also use inexact reduced FETI-DP and solve the linear system $(DK)X = L$ instead. This equates to the solution of one linear system with several right hand sides and thus we have 9 additional solves in 3D and 4 additional solves in 2D. This can be an expensive step and, in extreme cases, the computation of the consistent tangent moduli can take up to 40% of the total time to solution. A less expensive approximate solution or even omitting the matrix $L^T\,(DK)^{-1}L$ leads to inexact tangent moduli and thus an inexact Newton method on the macroscale.

A discussion of different strategies might be interesting. However, we always computed the complete consistent tangent modulus as suggested in [128] in order to ensure a stable convergence.
Finally, we use collective communication to provide $\overline{\mathbb{A}}^h$ and $\overline{P}^h$ on all MPI-ranks in order to assemble the linearized macroscopic problem.

5.3 Numerical Results

We present scaling results for the computational scale bridging using the FE^2 method in 2D and 3D. For the first time, scalability to 458 752 cores of the JUQUEEN BlueGene/Q at Forschungszentrum Jülich is achieved for our approach. All results presented here have been obtained during the ***JUQUEEN Extreme Scaling Workshop 2015***; see [17, 79].
As a phenomenological material law on the microscale we choose Neo-Hooke material; see Section 3.4.1 or equation (5.2) for the definition. As a macroscopic model problem, we consider a cube clamped on two opposite sides. We then pull apart both sides and extend the cube by 1%. As RVE, we consider a cubic domain with inclusions of random shape and size. The inclusions are thus local with respect to the FETI-DP subdomains. We consider a stiff material with an elasticity modulus of $E = 210000$ in the inclusions and softer material with an elasticity modulus of $E = 210$ in the surrounding material. We again choose Poisson's ratio as $\nu = 0.3$ in the whole domain. This parameter set is a common choice for steel and the coefficient jump in E represents the different materials: martensite and ferrite. Of course, this is only a reasonable description within the elastic range of steel and thus only valid for small deformations. Computations with a plasticity model and a realistic microstructure are already planned for the near future; see Fig. 5.5 for the geometry of a realistic RVE and some preliminary test results using a elasticity-plasticity model.
An illustration of the heterogeneous microstructure used in our scaling tests can be found in Fig. 5.4.

In our weak scalability tests, we scale up the size of the macroscopic problem while keeping the size of the microscopic RVEs fixed. We also do not change the number of FETI-DP subdomains for each RVE fixed and use one MPI rank for each FETI-DP subdomain. As we increase the number of processor cores in proportion to the problem size (weak scalability), in the best case, for a parallel efficiency of 100%, we would expect a constant time to solution.

In Tables 5.1 and 5.2, we see weak scalability for 2D and 3D. There we use one MPI rank for each BG/Q processor core. The base line for our parallel

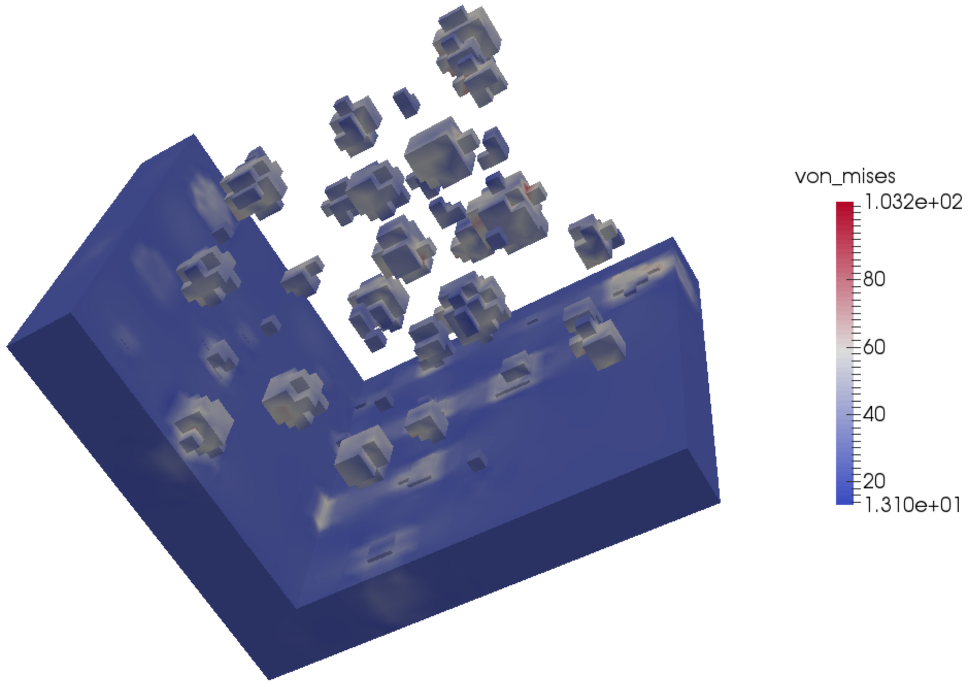

Fig. 5.4: Microscopic problem decomposed into $6 \times 6 \times 6$ subdomains with random inclusions local to the subdomains. We consider stiff material in the inclusions and softer hull material. Except for the L-shaped part, the hull material is removed to highlight the microscopic structure. We see a typical stress distribution with peaks caused by the microscopic structure.

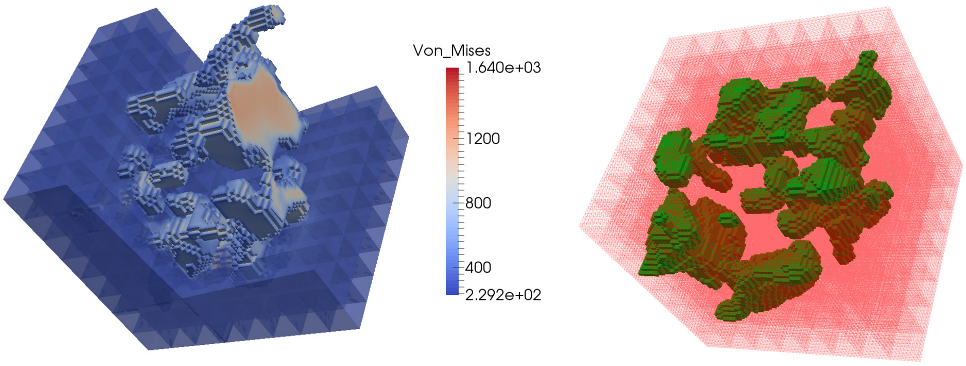

Fig. 5.5: Left: Stress plot of a preliminary test on one RVE, using a realistic elasticity-plasticity model. **Right:** Realistic representative volume element for modern high strength steels decomposed into $8 \times 8 \times 8$ subdomains.

efficiency is the smallest meaningful macroscopic problem, i.e., with 8 Gauß points in 2D and 16 Gauß points in 3D. A parallel efficiency of approximately 98% is achieved; see Tables 5.1 and 5.2. In Fig. 5.6 and Fig. 5.7 the data from these tables is depicted.

In Tables 5.1 and 5.2, the number of subdomains for each RVE, i.e., 256 in 2D and 512 in 3D, is still small. In Table 5.3, starting from the largest problem in Table 5.1, the size of the RVEs is increased by a factor of 4.

Next, we consider the effect of an overcommit. In Table 5.4, we show weak scaling but using an overcommit with up to 4 MPI ranks for each BlueGene/Q processor core. In the latter case, not more than 256 MB are available for each MPI rank. We use 16, 32, and 64 MPI ranks for each node and the RVE size is kept constant, i.e., the total problem size is increased by a factor of 4. We cannot expect perfect scalability in this situation. But we still see that acceptable scalability is obtained when scaling from a total of $458\,752$ MPI ranks to $917\,504$ MPI ranks, i.e., the total time to solution is $266.47s$ instead of $2 \cdot 215.41s = 430.82s$. Using $1\,835\,008$ MPI ranks does only result in small additional savings. This is partially caused by a comparable large macroscopic problem. The time spent for the solution of the macroscopic problem is increased by a factor of 8 when scaling from $917\,504$ MPI ranks to $1\,835\,008$ MPI ranks; see column *Time Macro. Solve* in Table 5.4, where the time of the macroscopic solves is summed up over all macroscopic Newton steps. This scalability bottleneck shows the need for a parallel solution method on the coarse level.

Let us conclude that our FE^{2}TI implementation shows nearly perfect scalability and scales easily up to the complete JUQUEEN. The inexact reduced FETI-DP methods perform well on the microscale and seem to be a good choice. Of course, for realistic simulations with a real microstructure on the RVE and considering plastic material behavior, we have to take care of the load balancing. In order to solve even larger macroscopic problems on larger machines, a parallelization of the macroscopic assembly and solve phases is necessary. The implementation should be straight forward.

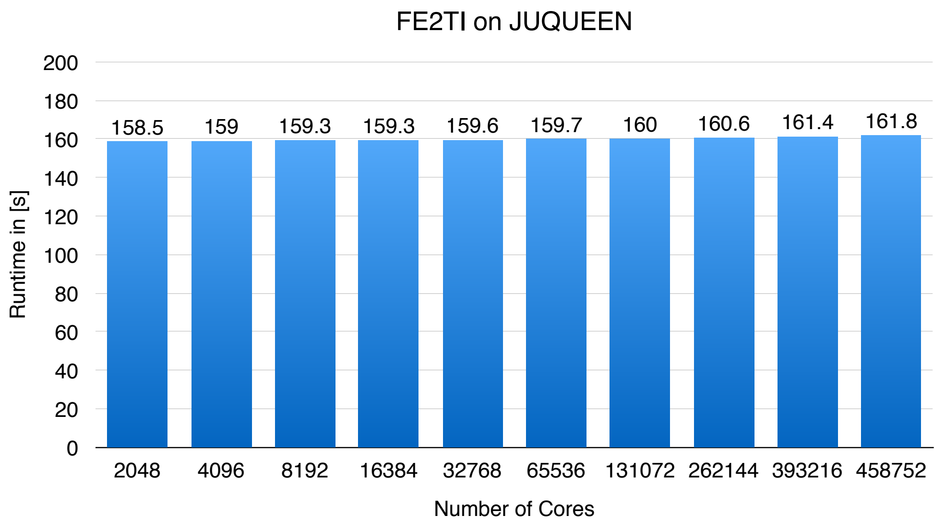

Fig. 5.6: FE^{2}TI: FE^2 in 2D using FETI-DP on each RVE. Weak scalability from 2 048 to 458 752 cores; data from Table 5.1; published in [79].

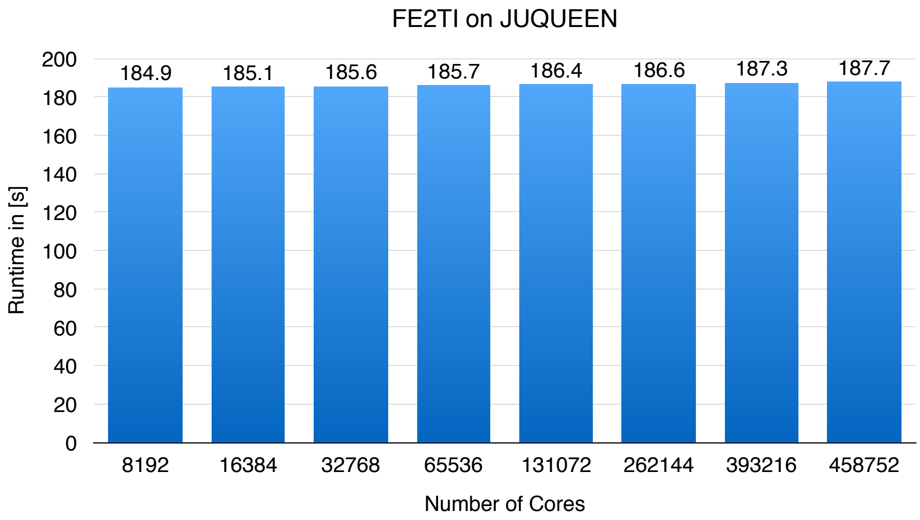

Fig. 5.7: FE^{2}TI: FE^2 in 3D using FETI-DP on each RVE. Weak scalability from 8 192 to 458 752 cores; data from Table 5.2; published in [79].

FE^{2}TI in 2D (Weak scaling)				
bg_size	**MPI-ranks**	**#RVEs**	**Time to Solution**	
128	2 048	8	158.47s	100.0%
256	4 096	16	159.03s	99.6%
512	8 192	32	159.27s	99.5%
1 024	16 384	64	159.32s	99.5%
2 048	32 768	128	159.58s	99.3%
4 096	65 536	256	159.68s	99.2%
8 192	131 072	512	159.99s	99.1%
16 384	262 144	1 024	160.62s	98.7%
24 576	393 216	1 536	161.41s	98.2%
28 672	458 752	1 792	161.78s	98.0%

Table 5.1: Scaling up the macro problem: FE^2 in 2D using FETI-DP on each RVE; heterogeneous hyperelasticity; P1 finite elements macro, P2 finite elements micro; 5.1 million d.o.f. on each RVE; 256 subdomains for each RVE; published in [79].

FE^{2}TI in 3D				
bg_size	**MPI-ranks**	**#RVEs**	**Time to Solution**	
512	8 192	16	184.86s	100.0%
1 024	16 384	32	185.09s	99.9%
2 048	32 768	64	185.61s	99.6%
4 096	65 536	128	185.72s	99.5%
8 192	131 072	256	186.43s	99.2%
16 384	262 144	512	186.61s	99.1%
24 576	393 216	768	187.32s	98.7%
28 672	458 752	896	187.65s	98.5%

Table 5.2: FE^2 in 3D using FETI-DP on each RVE; heterogeneous hyperelasticity; Q1 finite elements macro, P2 finite elements micro; 1.6m d.o.f. on each RVE; 512 subdomains for each RVE; published in [79].

FE^{2}TI in 2D (Increasing RVE sizes)					
bg_size	**MPI-ranks**	**#RVEs**	**RVE-size**	**RVE-size × #RVEs**	**Time to Solution**
28 672	458 752	1 792	5 126 402	9 186 512 384	161.78s
28 672	458 752	1 792	7 380 482	13 225 823 744	248.19s
28 672	458 752	1 792	13 117 442	23 506 456 064	483.68s
28 672	458 752	1 792	20 492 802	36 723 101 184	817.06s

Table 5.3: We increase the RVE sizes starting from the largest problem in Table 5.1; heterogeneous hyperelasticity; P1 finite elements macro, P2 finite elements micro; published in [79].

FE^{2}TI in 3D (1x, 2x, 4x MPI overcommit)						
	ranks	**MPI**		**Time**	**Time to**	
bg_size	**per node**	**ranks**	**#RVEs**	**Macro. Solves**	**Solution**	**Eff.**
28 672	16	458 752	896	1.53s	215.41s	100%
28 672	32	917 504	1 792	5.31s	266.47s	81%
28 672	64	1 835 008	3 584	43.20s	522.10s	41%

Table 5.4: Weak scaling efficiency using 16 / 32 / 64 MPI-ranks per node. FE^2 in 3D using FETI-DP on each RVE. Here, due to the memory constraints, we use 1 594 323 d.o.f. per RVE and 512 subdomains per RVE; published in [79].

6 Conclusion and Future Work

6.1 Conclusion

In this thesis, we have described new nonlinear and nonoverlapping domain decomposition methods of FETI-DP and BDDC type. These methods have shown the ability to reduce the number of Krylov steps and increase the ratio between local work and global work. Thus, the need for communication and synchronization is reduced. We have seen that the choice of the coarse space directly influences the convergence of the Newton method, which is different to standard Newton-Krylov approaches. The algorithmic building blocks are largely identical to classical FETI-DP or BDDC methods and in a nonlinear FETI-DP implementation these blocks just have to be rearranged.

We presented a highly scalable implementation of the irNonlinear-FETI-DP-1 approach, combining the nonlinear FETI-DP framework and the inexact reduced FETI-DP method. In this method an AMG solver is incorporated on the FETI-DP coarse variables in order to increase scalability. We presented weak and strong scalability on up to 262 144 cores on different architectures. We also discussed the usage of different AMG approaches for systems of PDEs and elasticity problems. We have seen that the global matrix approach can optimize scalability of inexact nonlinear FETI-DP methods applied to elasticity problems.

As a final application, we presented the FE^{2}TI approach. This combination of the computational scale bridging approach FE^2 and irNonlinear-FETI-DP-1 recently scaled up to the complete JUQUEEN BlueGene/Q at FZ Jülich.

6.2 Future Work

In order to overcome the scalability limits of our current inexact reduced Nonlinear-FETI-DP-1 implementation and to scale up to the largest machines, we will consider different improvements. First, running the AMG preconditioner of the FETI-DP coarse problem on a subcommunicator on separate cores could save memory on the remaining cores and thus increase the possible problem size. Second, we plan to incorporate an OpenMP parallel assembly and use a thread parallel sparse direct solver as PARDISO. We already presented some preliminary studies in [80].

In order to further customize AMG for FETI-DP in the context of elasticity and plasticity problems, further tests considering the GM approach in 3D with different material laws have to be performed. Also an investigation, which vectors have to be interpolated in the case of plasticity problems is necessary.

Also a parallel implementation of the inexact reduced Nonlinear-FETI-DP-2 approach suggested in this thesis would be interesting. In this method perfectly independent and local nonlinear problems have to be solved in order to compute the righthand side. Local line search methods could be considered and, if some of the independent problems converge fast, the corresponding cores will need new tasks, as, e.g., the acceleration of the computations on the remaining cores.

In general, in sophisticated elasticity and plasticity problems with heterogeneous materials, the standard FETI-DP coarse spaces with vertex and edge constraints might not suffice. Several approaches to compute an adaptive coarse space by solving localized generalized eigenvalue problems have been introduced in the recent years, as, e.g., [38, 39, 82, 106, 131]. An investigation of adaptive coarse spaces for nonlinear FETI-DP methods would be an interesting and very challenging task since proper coarse spaces can accelerate the convergence of the Newton method in the case of nonlinear FETI-DP.

Finally, we plan to perform some production runs for realistic steel deformation problems with our highly parallel FE^2 implementation in the near future.

Bibliography

[1] Cleve Ashcraft, Daniel Pierce, David K. Wah, and Jason Wu. The reference manual for SPOOLES, release 2.2: An object oriented software library for solving sparse linear systems of equations, 1999.

[2] Allison H. Baker, Robert D. Falgout, Tzanio V. Kolev, and Ulrike Meier Yang. Scaling hypres multigrid solvers to 100,000 cores. In Berry, Gallivan, Gallopoulos, Grama, Philippe, Saad, and Saied, editors, High-Performance Scientific Computing, pages 261–279. Springer London, 2012.

[3] Allison H. Baker, Tzanio V. Kolev, and Ulrike Meier Yang. Improving algebraic multigrid interpolation operators for linear elasticity problems. Numer. Linear Algebra Appl., 17(2-3):495–517, 2010.

[4] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, and Hong Zhang. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2014.

[5] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.5, Argonne National Laboratory, 2014.

[6] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry 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. Birkhauser Press, 1997.

[7] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework. Computing, 82(2-3):103–119, 2008.

[8] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and tests in DUNE. Computing, 82(2-3):121–138, 2008.

[9] Manoj Bhardwaj, David Day, Charbel Farhat, Michel Lesoinne, Kendall Pierson, and Daniel Rixen. Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems. Int. J. Numer. Meth. Eng., 47:513–535, 2000.

[10] Manoj Bhardwaj, Kendall H. Pierson, Garth Reese, Tim Walsh, David Day, Ken Alvin, James Peery, Charbel Farhat, and Michel Lesoinne. Salinas: A scalable software for high performance structural and mechanics simulation. In ACM/IEEE Proceedings of SC02: High Performance Networking and Computing. Gordon Bell Award, pages 1–19, 2002.

[11] Felipe Bordeu, Pierre-Alain Boucard, and Pierre Gosselet. Balancing domain decomposition with nonlinear relocalization: Parallel implementation for laminates. In B.H.V. Topping and P. Iványi, editors, Proceedings of the First International Conference on Parallel, Distributed and Grid Computing for Engineering, Stirlingshire, UK, 2009. Civil-Comp Press.

[12] Dietrich Braess. Towards algebraic multigrid for elliptic problems of second order. Computing, 55(4):379–393, 1995.

[13] Dietrich Braess. Finite Elemente, volume 4. Springer, Berlin, 2007.

[14] W. A. Marcel Brekelmans, Varvara Kouznetsova, and Marc G. D. Geers. Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme. Int. J. Numer. Meth. Eng., 54(8):1235–1260, 2002.

[15] Marian Brezina, Andy J. Cleary, Rob D. Falgout, Jim E. Jones, Tom A. Manteufel, Steve F. McCormick, and John W. Ruge. Algebraic multigrid based on element interpolation (AMGe). SIAM J. Sci. Comput., 22:1570–1592, 2000. Also available as LLNL technical report UCRL-JC-131752.

[16] Marian Brezina, Charles Tong, and Richard Becker. Parallel algebraic multigrid methods for structural mechanics. SIAM J. Sci. Comput., 27(5):1534–1554, 2006.

[17] Dirk Brömmel, Wolfgang Frings, and Brian J. N. Wylie. JUQUEEN extreme scaling workshop. Technical Report FZJ-JSC-IB-2015-01, Jülich Supercomputing Centre, 2015.

[18] Peter Brune, Matthew G. Knepley, Barry Smith, and Xuemin Tu. Composing scalable nonlinear algebraic solvers. Technical Report ANL/MCS-P2010-0112, Argonne National Laboratory, 2013.

[19] Marco Buck, Oleg Iliev, and Heiko Andrä. Multiscale finite elements for linear elasticity: Oscillatory boundary conditions. In Jocelyne Erhel, Martin J. Gander, Laurence Halpern, Géraldine Pichot, Taoufik Sassi, and Olof Widlund, editors, Domain Decomposition Methods in Science and Engineering XXI, volume 98 of Lecture Notes in Computational Science and Engineering, pages 237–245. Springer International Publishing, 2014.

[20] V. E. Bulgakov. Multi-level iterative technique and aggregation concept with semi-analytical preconditioning for solving boundary value problems. Comm. Numer. Methods Eng., 9(8):649–657, 1993.

[21] Xiao-Chuan Cai and Maksymilian Dryja. Domain decomposition methods for monotone nonlinear elliptic problems. In Domain decomposition methods in scientific and engineering computing (University Park, PA, 1993), volume 180 of Contemp. Math., pages 21–27. Amer. Math. Soc., Providence, RI, 1994.

[22] Xiao-Chuan Cai and David E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. SIAM J. Sci. Comput., 24(1):183–200 (electronic), 2002.

[23] Xiao-Chuan Cai, David E. Keyes, and Leszek Marcinkowski. Nonlinear additive Schwarz preconditioners and application in computational fluid dynamics. Internat. J. Numer. Methods Fluids, 40(12):1463–1470, 2002. LMS Workshop on Domain Decomposition Methods in Fluid Mechanics (London, 2001).

[24] Tanja Clees. AMG Strategies for ODE Systems with Applications in Industrial Semiconductor Simulation. Shaker Verlag GmbH, Germany, 2005.

[25] Philippe Cresta, Olivier Allix, Christian Rey, and Stéphane Guinard. Nonlinear localization strategies for domain decomposition methods: application to post-buckling analyses. Comput. Methods Appl. Mech. Eng., 196(8):1436–1446, 2007.

[26] Jean-Michel 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] Timothy A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw., 30(2):196–199, June 2004.

[28] Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw., 30(2):165–195, June 2004.

[29] Hans De Sterck, Robert D. Falgout, Joshua W. Nolting, and Ulrike Meier Yang. Distance-two interpolation for parallel algebraic multigrid. Numer. Linear Algebra Appl., 15:115–139, 2008.

[30] Hans De Sterck, Ulrike Meier Yang, and Jeffrey J. Heys. Reducing complexity in parallel algebraic multigrid preconditioners. SIAM J. Matrix Anal. Appl., 27(4):1019–1039 (electronic), 2006.

[31] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. SIAM J. Matrix Analysis and Applications, 20(3):720–755, 1999.

[32] Simone Deparis. Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation. PhD thesis, EPFL, 2004.

[33] Simone Deparis, Marco Discacciati, Gilles Fourestey, and Alfio Quarteroni. Heterogeneous Domain Decomposition Methods for Fluid-Structure Interaction Problems, volume 55 of Domain Decomposition Methods in Science and Engineering XVI. Springer Berlin Heidelberg, 2005.

[34] Simone Deparis, Marco Discacciati, Gilles Fourestey, and Alfio Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. Comput. Methods Appl. Mech. Eng., 195:5797–5812, 2006.

[35] F. Devries, H. Dumontet, G. Duvaut, and F. Lene. Homogenization and damage for composite structures. Int. J. Numer. Meth. Eng., 27(2):285–298, 1989.

[36] Clark R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput., 25(1):246–258, 2003.

[37] Clark R. Dohrmann. Interpolation operators for algebraic multigrid by local optimization. SIAM J. Sci. Comput., 29(5):2045–2058 (electronic), 2007.

[38] Victorita Dolean, Patrice Hauret, Frederic Nataf, Clemens Pechstein, Robert Scheichl, and Nicole Spillane. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. NuMa-Report, Institute of Computational Mathematics, Johannes Kepler University Linz, 7, 2011.

[39] Victorita Dolean, Frédéric Nataf, Robert Scheichl, and Nicole Spillane. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. Comput. Meth. in Appl. Math., 12(4):391–414, 2012.

[40] Maksymilian Dryja and Wolfgang Hackbusch. On the nonlinear domain decomposition method. BIT, 37(2):296–311, 1997.

[41] Maksymilian Dryja and Olof B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. Comm. Pure Appl. Math, 48:121–155, 1995.

[42] Robert D. Falgout, Jim E. Jones, and Ulrike Meier Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. In AreMagnus Bruaset and Aslak Tveito, editors, Numerical

Solution of Partial Differential Equations on Parallel Computers, volume 51 of Lecture Notes in Computational Science and Engineering, pages 267–294. Springer Berlin Heidelberg, 2006.

[43] Charbel Farhat, Michel Lesoinne, Patrick LeTallec, Kendall Pierson, and Daniel Rixen. FETI-DP: A Dual-Primal Unified FETI Method - Part I: A Faster Alternative to the Two-Level FETI Method. Internat. J. Numer. Methods Eng., 50:1523–1544, 2001.

[44] Charbel Farhat, Michel Lesoinne, and Kendall Pierson. A scalable dual-primal domain decomposition method. Numer. Lin. Alg. Appl., 7:687–714, 2000.

[45] Charbel Farhat and Jan Mandel. The two-level FETI method for static and dynamic plate problems - part I: an optimal iterative solver for biharmonic systems. Comp. Meth. Appl. Mech. Eng., 155:129–152, 1998.

[46] Charbel Farhat, Jan Mandel, and Francois-Xavier Roux. Optimal convergence properties of the FETI domain decomposition method. Comput. Methods Appl. Mech. Eng., 115:367–388, 1994.

[47] Charbel Farhat, Kendall Pierson, and Michel Lesoinne. The second generation of FETI methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems. Comp. Meth. Appl. Mech. Eng., 184:333–374, 2000.

[48] Miguel Ángel Fernández, Jean-Frédéric Gerbeau, Antoine Gloria, and Marina Vidrascu. Domain decomposition based Newton methods for fluid-structure interaction problems. In CANUM 2006—Congrès National d'Analyse Numérique, volume 22 of ESAIM Proc., pages 67–82. EDP Sci., Les Ulis, 2008.

[49] Frédéric Feyel. Multiscale FE^2 elastoviscoplastic analysis of composite structures. Comp. Mat. Science, 16:344–354, 1999.

[50] Frédéric Feyel and Jean-Louis Chaboche. FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. Comp. Meth. Appl. Mech. Eng., 183(3):309 – 330, 2000.

[51] Jacob Fish, Kamlun Shek, Muralidharan Pandheeradi, and Mark S. Shephard. Computational plasticity for composite structures based on mathematical homogenization: Theory and practice. Comp. Meth. Appl. Mech. Eng., 148(1-2):53 – 73, 1997.

[52] Gilles A. Francfort. Homogenization and linear thermoelasticity. SIAM J. Math. Anal., 14(4):696–708, 1983.

[53] Benjamin Ganis, Kundan Kumar, Gergina Pencheva, Mary F. Wheeler, and Ivan Yotov. A global Jacobian method for mortar discretizations of a fully-implicit two-phase flow model. SIAM Multiscale Model. Simul., 12(4):1401–1423, 2014.

[54] Benjamin Ganis, Gergina Pencheva, Mary F. Wheeler, Tim Wildey, and Ivan Yotov. A frozen Jacobian multiscale mortar preconditioner for nonlinear interface operators. Multiscale Model. Simul., 10(3):853–873, 2012.

[55] Alan George. Nested dissection of a regular finite element mesh. SIAM J. Numer. Anal., 10:345–363, 1973.

[56] Alan George and Joseph Liu. The evolution of the minimum degree ordering algorithm. SIAM Rev., 31:1–19, 1989.

[57] Sabrina Gippert. Domain decomposition methods for elastic materials with compressible and almost incompressible components. PhD thesis, 2012. Universität Duisburg-Essen, 2013.

[58] Sabrina Gippert, Axel Klawonn, and Oliver Rheinbach. Analysis of FETI-DP and BDDC for linear elasticity in 3D with almost incompressible components and varying coefficients inside subdomains. SIAM J. Numer. Anal., 50(5):2208 –2236, 2012.

[59] Sabrina Gippert, Axel Klawonn, and Oliver Rheinbach. A deflation based coarse space in dual-primal FETI methods for almost incompressible elasticity. In Assyr Abdulle, Simone Deparis, Daniel Kressner, Fabio Nobile, and Marco Picasso, editors, Numerical Mathematics and Advanced Applications - ENUMATH 2013, volume 103 of Lecture Notes in Computational Science and Engineering, pages 573–581. Springer International Publishing, 2015.

[60] Nicholas I. M. Gould, Jennifer A. Scott, and Yifan Hu. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. ACM Trans. Math. Softw., 33(2), June 2007.

[61] Anne Greenbaum. Iterative methods for solving linear systems, volume 17 of Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[62] Michael Griebel, Daniel Oeltz, and Alexander Schweitzer. An algebraic multigrid for linear elasticity. J. Sci. Comp., 25(2):385–407, 2003.

[63] Christian Gross and Rolf Krause. On the convergence of recursive trust-region methods for multiscale nonlinear optimization and applications to nonlinear mechanics. SIAM J. Numer. Anal., 47(4):3044–3069, 2009.

[64] Christian Groß and Rolf Krause. A generalized recursive trust-region approach - nonlinear multiplicatively preconditioned trust-region methods and applications. Technical Report 2010-09, Institute of Computational Science, Università della Svizzera italiana, 03 2010.

[65] Christian Groß and Rolf Krause. On the globalization of ASPIN employing trust-region control strategies - convergence analysis and numerical examples. Technical Report 2011-03, Institute of Computational Science, Università della Svizzera italiana, 01 2011.

[66] Alexander Heinlein, Ulrich Hetmaniuk, Axel Klawonn, and Oliver Rheinbach. The approximate component mode synthesis special finite element method in two dimensions: Parallel implementation and numerical results. J. Comp. Appl. Math., 2015.

[67] Van E. Henson and Panayot S. Vassilevski. Element-free AMGe: general algorithms for computing interpolation weights in AMG. SIAM J. Sci. Comput., 23(2):629–650 (electronic), 2001. Copper Mountain Conference (2000).

[68] Van E. Henson and Ulrike Meier Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. Appl. Numer. Math., 41:155–177, 2002.

[69] Magnus R. Hestenes. The conjugate gradient method for solving linear systems. In Proc. Symp. Appl. Math VI, American Mathematical Society, pages 83–102, New York, 1956. McGraw-Hill.

[70] Gerhard A. Holzapfel. Nonlinear Solid Mechanics. A Continuum Approach for Engineering. John Wiley and Sons, Chichester, 2000.

[71] Feng-Nan Hwang and Xiao-Chuan 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.

[72] Feng-Nan Hwang and Xiao-Chuan Cai. A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms. Comput. Methods Appl. Mech. Eng., 196(8):1603–1611, 2007.

[73] Olaf Ippisch, Markus Blatt, Jorrit Fahlke, and Felix Heimann. muphi - simulation of flow and transport in porous media, 2014. http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/muPhi/_node.html.

[74] C. Tim Kelley. Iterative methods for linear and nonlinear equations, volume 16 of Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. With separately available software.

[75] Axel Klawonn, Martin Lanser, Patrick Radtke, and Oliver Rheinbach. On an adaptive coarse space and on nonlinear domain decomposition. In Jocelyne Erhel, Martin J. Gander, Laurence Halpern, Géraldine Pichot, Taoufik Sassi, and Olof B. Widlund, editors, Domain Decomposition Methods in Science and Engineering XXI, volume 98 of Lect. Notes Comput. Sci. Eng., pages 71–83. Springer-Verlag, 2014.

[76] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Nonlinear FETI-DP and BDDC methods. SIAM J. Sci. Comput., 36(2):A737–A765, 2014.

[77] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. A nonlinear FETI-DP method with an inexact coarse problem. 2014. Accepted to the

proceedings of the 22nd International Conference on Domain Decomposition Methods, Sept. 16-20, 2013, Università della Svizzera italiana, Lugano, Switzerland. Preprint: http://www.mathe.tu-freiberg.de/files/personal/253/rheinbach-plenarytalk-dd22.pdf.

[78] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Towards extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations. Technical Report 2014-13, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 2014. Submitted to Siam J. Sci. Comput., http://tu-freiberg.de/fakult1/forschung/preprints.

[79] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. EXASTEEL - computational scale bridging using a FE^{2}TI approach with ex_nl/fe^2. In Dirk Brömmel, Wolfgang Frings, and Brian J. N. Wylie, editors, JUQUEEN Extreme Scaling Workshop 2015, pages 15–21, 2015.

[80] Axel Klawonn, Martin Lanser, Oliver Rheinbach, Holger Stengel, and Gerhard Wellein. Hybrid MPI/OpenMP parallelization in FETI-DP methods. Technical Report 2015-02, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 2015. Accepted for publication in Springer Lect. Notes Comput. Sci. Eng., http://tu-freiberg.de/fakult1/forschung/preprints.

[81] Axel Klawonn, Luca F. Pavarino, and Oliver Rheinbach. Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains. Comput. Meth. Appl. Mech. Eng., 198:511–523, 2008.

[82] Axel Klawonn, Patrick Radtke, and Oliver Rheinbach. FETI-DP methods with an adaptive coarse space. SIAM J. Numer. Anal., 53(1):297 – 320, 2015.

[83] Axel Klawonn and Oliver Rheinbach. A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis. SIAM J. Sci. Comput., 28(5):1886–1906, 2006.

[84] Axel Klawonn and Oliver Rheinbach. Inexact FETI-DP methods. Internat. J. Numer. Methods Eng., 69(2):284–307, 2007.

[85] Axel Klawonn and Oliver Rheinbach. Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. Comput. Methods Appl. Mech. Eng., 196(8):1400–1414, 2007.

[86] Axel Klawonn and Oliver Rheinbach. A hybrid approach to 3-level FETI. PAMM, pages 10841–10843, 2008. DOI: 10.1002/pamm.200810841, Special issue: 79th annual meeting of the International Association of Applied Mathematics and Mechanics (GAMM).

[87] Axel Klawonn and Oliver Rheinbach. Highly scalable parallel domain decomposition methods with an application to biomechanics. ZAMM Z. Angew. Math. Mech., 90(1):5–32, 2010.

[88] Axel Klawonn and Oliver Rheinbach. Deflation, projector preconditioning, and balancing in iterative substructuring methods: Connections and new results. SIAM J. Sci. Comp., 34(1):459–484, 2012.

[89] Axel Klawonn, Oliver Rheinbach, and Martin Lanser. FE2TI (ex_nl/FE2) - exasteel - bridging scales for multiphase steels, 2015. http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/FE2TI/_node.html.

[90] Axel Klawonn, Oliver Rheinbach, and Olof B. Widlund. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. SIAM J. Numer. Anal., 46(5):2484–2504, 2008.

[91] Axel Klawonn, Matthias Uran, Martin Lanser, and Oliver Rheinbach. An inexact version of a nonlinear FETI-DP method. Technical report, 2015. Preprint in preparation.

[92] Axel Klawonn and Olof B. Widlund. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. Comm. Pure Appl. Math., LIV:57–90, 2001.

[93] Axel Klawonn and Olof B. Widlund. Dual-Primal FETI Methods for Linear Elasticity. Comm. Pure Appl. Math., 59(11):1523–1572, 2006.

[94] Axel Klawonn, Olof B. Widlund, and Maksymilian Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. SIAM J. Numer. Anal., 40(1):159–179, 2002.

[95] Dana A. Knoll and David E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J. Comput. Phys., 193(2):357–397, 2004.

[96] Varvara Kouznetsova, W. A. Marcel Brekelmans, and Frank P. T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. Comp. Mech., 27:37–48, 2001.

[97] Andrey Kuzmin, Mathieu Luisier, and Olaf Schenk. Fast methods for computing selected elements of the greens function in massively parallel nanoelectronic device simulations. In F. Wolf, B. Mohr, and D. Mey, editors, Euro-Par 2013 Parallel Processing, volume 8097 of Lecture Notes in Computer Science, pages 533–544. Springer Berlin Heidelberg, 2013.

[98] Patrick Le Tallec. Numerical homogeneisation technique with domain decomposition based a-posteriori error estimates. In Michel Bercovier, MartinJ. Gander, Ralf Kornhuber, and Olof Widlund, editors, Domain Decomposition Methods in Science and Engineering XVIII, volume 70 of Lecture Notes in Computational Science and Engineering, pages 27–37. Springer Berlin Heidelberg, 2009.

[99] Jing Li and Olof B. Widlund. FETI–DP, BDDC, and Block Cholesky Methods. Internat. J. Numer. Methods Eng., 66(2):250–271, 2006.

[100] Xiaoye S. Li. An overview of SuperLU: Algorithms, implementation, and user interface. ACM Trans. Math. Software, 31(3):302–325, September 2005.

[101] Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. SIAM J. Numer. Anal., 16(2):346–358, 1979.

[102] Oriol Lloberas-Valls, Daniel Rixen, Angelo Simone, and Lambertus J. Sluys. Domain decomposition techniques for the efficient modeling of brittle heterogeneous materials. Comp. Meth. Appl. Mech. Eng., 200(13-16):1577 – 1590, 2011.

[103] Jan Mandel. Balancing domain decomposition. Comm. Numer. Meth. Eng., 9:233–241, 1993.

[104] Jan Mandel and Clark R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. Numer. Linear Algebra Appl., 10:639–659, 2003.

[105] Jan Mandel, Clark R. Dohrmann, and Radek Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. Appl. Numer. Math., 54:167–193, 2005.

[106] Jan Mandel and Bedřich Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. Comp. Meth. Appl. Mech. Eng., 196(8):1389–1399, 2007.

[107] Jan Mandel, Bedřich Sousedík, and ClarkR. Dohrmann. On Multilevel BDDC. In Ulrich Langer, Marco Discacciati, David E. Keyes, Olof B. Widlund, and Walter Zulehner, editors, Domain Decomposition Methods in Science and Engineering XVII, volume 60 of Lecture Notes in Computational Science and Engineering, pages 287–294. Springer Berlin Heidelberg, 2008.

[108] Jan Mandel and Radek Tezaur. On the convergence of a dual-primal substructuring method. Numer. Math., 88:543–558, 2001.

[109] MATLAB. version 7.13.0.564 (R2011a). The MathWorks Inc., Natick, Massachusetts, 2011.

[110] Christian Miehe, Jörg Schröder, and Jan Schotte. Computational homogenization analysis in finite plasticity. Simulation of texture development in polycrystalline materials. Comp. Meth. Appl. Mech. Eng., 171:387–418, 1999.

[111] Adrian C. Muresan and Yvan Notay. Analysis of aggregation-based multigrid. SIAM J. Sci. Comput., 30:1082–1103, 2008.

[112] Jorge Nocedal and Stephen J. Wright. Numerical optimization. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[113] Yvan Notay. An aggregation-based algebraic multigrid method. Electron. Trans. Numer. Anal., 37:123–146, 2010.

[114] Yvan Notay and Artem Napov. Algebraic analysis of aggregation-based multigrid. Numer. Linear Algebra Appl., 18:539–564, 2011.

[115] J. Tinsley Oden, Kumar Vemaganti, and Nicolas Moës. Hierarchical modeling of heterogeneous solids. Comp. Meth. Appl. Mech. Eng., 172(1-4):3–25, 1999.

[116] James M. Ortega and Werner 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.

[117] Julien Pebrel, Christian Rey, and Pierre Gosselet. A nonlinear dual-domain decomposition method: Application to structural problems with damage. Inter. J. Multiscal Comp. Eng., 6(3):251–262, 2008.

[118] Oliver Rheinbach. Parallel iterative substructuring in structural mechanics. Arch. Comput. Methods Eng., 16(4):425–463, 2009.

[119] Oliver Rheinbach, Axel Klawonn, Jörg Schröder, Gerhard Wellein, and Daniel Balzani. Exasteel - bridging scales for multiphase steels, 2014. http://www.sppexa.de/general-information/projects.html#EXASTEEL.

[120] Ulrich Rüde. Terra-neo - integrated co-design of an exascale earth mantle modeling framework, 2014. http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/Terra-Neo/_node.html.

[121] John W. Ruge. AMG for problems of elasticity. Appl. Math. Comp., 19:293–309, 1986.

[122] John W. Ruge and Klaus Stüben. Algebraic multigrid (AMG). Multigrid Methods, 3, 1987.

[123] Yousef Saad. Iterative Methods for Sparse Linear Systems. PWS Publishing Company, 1996.

[124] Yousef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comp., 7:856–869, 1986.

[125] Olaf Schenk, Matthias Bollhöfer, and Rudolf A. Römer. On large-scale diagonalization techniques for the anderson model of localization. SIAM Rev., 50(1):91–112, February 2008.

[126] Olaf Schenk, Andreas Wächter, and Michael Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. Comp. Opt. Appl., 36(2-3):321–341, 2007.

[127] Jörg Schröder. Homogenisierungsmethoden der nichtlinearen Kontinuumsmechanik unter Beachtung von Stabilitätsproblemen. PhD thesis, Bericht aus der Forschungsreihe des Institut für Mechanik (Bauwesen), Lehrstuhl I, 2000. Habilitationsschrift.

[128] Jörg Schröder and Klaus Hackl, editors. Plasticity and beyond: microstructures, crystal-plasticity and phase transitions, volume 550 of CISM Courses and Lectures. Springer, Vienna, 2014. Lectures from the CISM Course held in Udine, June 27–July 1, 2011.

[129] R.J.M. Smit, W.A.Marcel Brekelmans, and Han E.H. Meijer. Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling. Comp. Meth. Appl. Mech. Eng., 155:181–192, 1998.

[130] Bedřich Sousedík and Jan Mandel. On adaptive-multilevel BDDC. In Domain decomposition methods in science and engineering XIX, volume 78 of Lect. Notes Comput. Sci. Eng., pages 39–50. Springer, Heidelberg, 2011.

[131] Nicole Spillane and Daniel J. Rixen. Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. Int. J. Numer. Meth. Eng., 95:953–990, 2013.

[132] Klaus Stüben. An introduction to algebraic multigrid. In Multigrid, pages 413–532. Academic Press, London, San Diego, 2001. Also available as GMD Report 70, November 1999.

[133] Andrea Toselli and Olof Widlund. Domain Decomposition Methods - Algorithms and Theory, volume 34 of Springer Series in Computational Mathematics. Springer, 2004.

[134] Xuemin Tu. Three-level BDDC in three dimensions. SIAM J. Sci. Comput., 29(4):1759–1780 (electronic), 2007.

[135] Xuemin Tu. Three-level BDDC in two dimensions. Inter. J. Numer. Meth. Eng., 69:33–59, 2007.

[136] Petr Vaněk, Jan Mandel, and Marian Brezina. Algebraic multigrid by smooth aggregation for second and fourth order elliptic problems. Computing, 56:179–196, 1996.

[137] Ulrike Meier Yang. Parallel algebraic multigrid methods - high performance preconditioners. In A.M. Bruaset and A. Tveito, editors, Numerical Solutions of Partial Differential Equations on Parallel Computers, pages 209–236. Springer-Verlag, Lecture Notes in Computational Science and Engineering, 2006.

[138] Ulrike Meier Yang. On long-range interpolation operators for aggressive coarsening. Numer. Linear Algebra Appl., 17:453–472, 2010.

[139] Olgierd C. Zienkiewicz and Robert L. Taylor. The Finite Element Method for Solid and Structural Mechanics. Elsevier, Oxford, 2005.

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, Patrick Radtke, and Oliver Rheinbach. On an adaptive coarse space and on nonlinear domain decomposition. In Jocelyne Erhel, Martin J. Gander, Laurence Halpern, Géraldine Pichot, Taoufik Sassi, and Olof B. Widlund, editors, Domain Decomposition Methods in Science and Engineering XXI, volume 98 of Lect. Notes Comput. Sci. Eng., pages 71-83. Springer-Verlag, 2014.

- Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Nonlinear FETI-DP and BDDC methods. SIAM J. Sci. Comput., 36(2):A737-A765, 2014.

- Axel Klawonn, Martin Lanser, and Oliver Rheinbach. A nonlinear FETI-DP method with an inexact coarse problem. 2014. Accepted to the proceedings of the 22nd International Conference on Domain Decomposition Methods, Sept. 16-20, 2013, Università della Svizzeria italiana, Lugano, Switzerland.

- Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Towards extremely scalable nonlinear domain decomposition methods for elliptical partial differetial equations. Technical Report 2014-13, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 2014. Submitted to SIAM J. Sci. Comput.

- Axel Klawonn, Martin Lanser, and Oliver Rheinbach. EXASTEEL - Computational Scale Bridging using a FE^{2}TI approach with ex_nl/fe^2. In Dirk Brömmel, Wolfgang Frings, and Brian J. N. Wylie, editor, JUQUEEN Extreme Scaling Workshop 2015, Technical Report, 2015.

Köln, den 21.04.2015 (Martin Lanser)