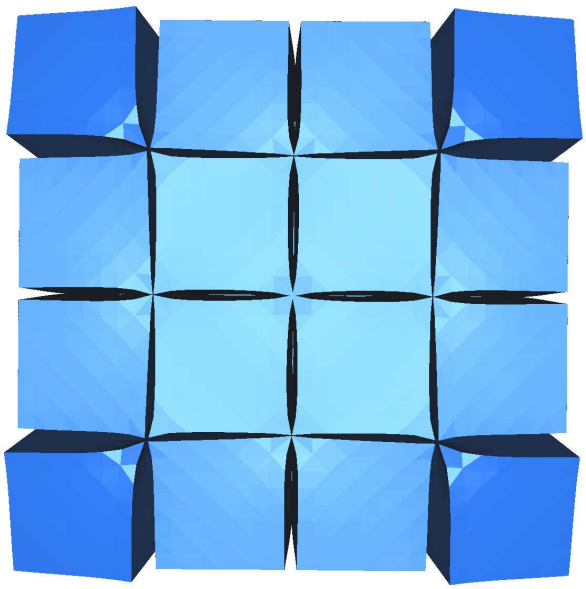

Parallel Scalable Iterative Substructuring: Robust Exact and Inexact FETI-DP Methods with Applications to Elasticity

Oliver Rheinbach

geboren in Hilden

Fachbereich Mathematik
Universität Duisburg-Essen

29. Mai 2006

Dissertation im Fach Mathematik
zum Erwerb des Dr. rer. nat.
am Fachbereich Mathematik
der Universität Duisburg-Essen

Erstgutachter: Prof. Dr. Axel Klawonn,
Fachbereich Mathematik, Universität Duisburg-Essen
Zweitgutachter: Prof. Dr. Olof B. Widlund,
Courant Institute of Mathematical Sciences, New York University
Drittgutachter: Prof. Dr. Gerhard Starke,
Institut für Angewandte Mathematik, Universität Hannover
Tag der mündlichen Prüfung: 13.07.2006

Contents

Acknowledgments

I would like to thank my advisor Axel Klawonn for the time and work devoted to me and to this thesis as well as for the encouragement he gave me. I am also grateful to Olof Widlund for inspiration and encouragement. I would like to thank both and Gerhard Starke for reviewing this work. Furthermore, I am happy to thank my colleagues at the University of Duisburg-Essen, especially Thomas Schamberg, with whom I have shared an office for many years. I would not have been able to finish this work without the support of my parents and my friends.

I am thankful for many fruitful discussions with Barry Smith, Satish Balay, Matthew Knepley, Jason Sarich and the PETSc team, while visiting the Mathematics and Computer Science Division at Argonne National Laboratory, USA. I would also like to thank Panayot Vassilevski, Ulrike Meier Yang, the hypre team and David Keyes at Lawrence Livermore National Laboratory.

I am very happy to express my thanks to Barbara Wohlmuth and Luca Pavarino as well as Andrea Toselli and Xavier Vasseur. I enjoyed working with them.

I am grateful to Tanja Clees, Klaus Stüben, and the Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Sankt Augustin, Germany, for providing SAMG and also wish to thank Hans-Jürgen Philipsenburg, Holger Hein and GETRAG FORD Transmissions GmbH, Cologne, Germany.

This work gratefully acknowledges the X-ray computer tomography cross sections which were obtained from Matthias Epple, Institute of Inorganic Chemistry, University of Duisburg-Essen, within a common research collaboration. Furthermore, this work gratefully acknowledges the use of Jazz, a 350 node computing cluster operated by the Mathematics and Computer Science Division at Argonne National Laboratory, USA, as part of its Laboratory Computing Resource Center. It also gratefully acknowledges the use of the 2 304 processor MCR cluster at the Lawrence Livermore National Laboratory, USA. Moreover, for this work computing time was used on the 96 processor Zivcluster of the Universität Münster, Germany, as part of the Rechnerverbund NRW. Additional testing was done, to some extent, on the 256 processor Rubens cluster of the Universität Siegen, the 1 024 processor ALiCEnext cluster of the Universität Wuppertal, and on our own 16 processor cluster in Essen. Many images in this work were rendered using Medit [48].

Chapter 1

Introduction

Many complex problems in physics and engineering are described by Partial Differential Equations (PDEs). In general, it is impossible to find exact solutions to these equations, and instead a discrete problem is defined, which represents an approximation of the original problem. Such a discrete problem can be defined, e.g., by using the finite element method (FEM). The discretization then results in large linear equation systems, which today can be of the order of 10^5 to 10^9 unknowns. These linear systems are often too large to be solved exactly. Domain decomposition methods are iterative algorithms to find approximate solutions for the large systems obtained from the discretization of Partial Differential Equations. They make use of underlying properties of the PDEs to achieve fast convergence by applying a hierarchical approach. In domain decomposition methods, the domain associated with the partial differential equation is decomposed into a, possibly large, number of subdomains. In these subdomains, local problems are defined, which are solved in each iteration step in order to define an approximate inverse of the system matrix. In order to obtain a numerical scalable algorithm, also a small coarse problem has to be introduced and solved in each iteration step. The divide-and-conquer approach taken in domain decomposition methods makes them particularly suitable for parallel computing.

In this work, we consider nonoverlapping domain decomposition methods belonging to the Dual-Primal Finite Element Tearing and Interconnecting (FETI-DP) methods, originally introduced by Farhat [40]. We will also consider its primal counterpart, the Balancing Domain Decomposition (BDDC) method due to Dohrmann [29], although our presentation will be biased towards the FETI-DP method.

We will present FETI-DP algorithms, along with parallel implementations, for compressible homogeneous and heterogeneous 3D linear elasticity, as well as scalar elliptic problems, for almost incompressible elasticity and

for higher order methods, i.e. spectral and *hp* finite elements.

Today's distributed memory supercomputers can have up to 10^5 processors. The design of algorithms that can make efficient use of such machines is a challenging task. The standard FETI-DP method is limited in parallel scalability almost exclusively by the exact solution of its coarse problem, see discussion below. We propose a new algorithmic variant, together with a family of inexact FETI-DP methods, which allows for an inexact solution of the FETI-DP coarse problem, and thus remove this limitation of the standard FETI-DP method.

The FETI-DP method was originally introduced in Farhat et al. [40] using vertex constraints and extended to 3D in Farhat, Lesoinne, and Pierson [41] using additional, optional average constraints. In Mandel and Tezaur [100], a convergence bound for 2D was provided. Then, in Klawonn and Widlund [84], Klawonn, Widlund, and Dryja [86, 87], and Klawonn and Rheinbach [76] a family of FETI-DP algorithms for 3D together with convergence bounds was introduced.

The family of FETI-DP algorithms descended from the earlier one-level and two-level FETI algorithms, see Farhat and Roux [45, 46], Farhat, Mandel, and Roux [43], Farhat and Mandel [42], Farhat, Pierson, and Lesoinne [44]. Already the FETI method has been used in large scale parallel simulations, e.g. [13]. The Dirichlet preconditioner without scaling was first introduced in Farhat, Mandel, and Roux [43]. For early references to the scaled version, see, e.g., Rixen and Farhat [111] and Klawonn and Widlund [83]. Let us note that appropriate scaling is important to obtain convergence results which are independent of jumps in the coefficients of the partial differential equation [83, 86, 84]. But also for homogeneous problems, scaling can be important to improve convergence and also in order to obtain an improved condition number bound, see Mandel and Tezaur [99] and Klawonn and Widlund [83].

FETI methods belong to the larger family domain decomposition methods without overlap, also referred to as iterative substructuring methods; for an introduction, see Smith, Bjørstad, and Gropp [127] and Toselli and Widlund [136].

In FETI-DP methods the continuity of the solution across the subdomain boundaries is enforced by Lagrange multipliers. This results in a mixed linear system with primal variables and Lagrange multipliers as unknowns. The basic idea of FETI-DP domain decomposition methods is to form a Schur complement by eliminating the primal variables and then iterate on the Lagrange multiplier variables, usually in combination with a preconditioner. Special attention has to be given to the elimination process of the primal variables since the associated local matrices are usually only semidefinite even though the overall mixed linear system is uniquely solvable. This is

due to local stiffness matrices belonging to subdomains lacking sufficient essential boundary conditions. In FETI-DP methods, a sufficient number of constraints, e.g. continuity constraints across the interface at selected nodes on the subdomain boundaries, is chosen such that the local stiffness matrices become invertible. We note that nodal constraints work well only in two dimensions, more elaborate choices, e.g. averages over edges or faces, are used in three dimensions in order to obtain a good convergence estimate, see Chapters 2 and 3. These additional primal constraints introduce a certain coupling between the otherwise completely decoupled local subdomain problems. This coupling also builds the coarse problem needed for scalability of the algorithm.

FETI-DP methods obtain their numerical and parallel scalability from the fact that the coarse problem is very small compared to the overall problem. This coarse problem is traditionally solved exactly by the use of a direct solver. Nevertheless, if a very large number of subdomains is used or if the problem requires the use of a larger coarse space, the cost of solving the coarse problem directly may become high. In standard FETI-DP methods, an inexact solution of the coarse problem is not straightforward since the coarse problem is, by means of the elimination process, built into the FETI-DP system matrix. Thus, an inexact solution in the elimination process of the primal variables would lead to a different linear system to be solved and thus to a perturbed solution, different to that of the original problem.

Another class of nonoverlapping domain decomposition methods, which is closely related to the FETI-DP algorithms, are the Balancing Domain Decomposition methods by Constraints (BDDC); see Cros [26], Dohrmann [29], Mandel and Dohrmann [97], Mandel, Dohrmann, and Tezaur [98], or Li and Widlund [92].

We note that there are also FETI methods using two sets Lagrange multipliers; see, e.g., Magoulès, Roux, and Salmon [95], Magoulès, Roux, and Series [96], or Series, Feyel, and Roux [121]; there are also other domain decomposition methods using Robin boundary conditions at the interface; see, e.g., Flauraud and Nataf [47] and the references given therein.

In Chapter 1 we have collected background material that may be helpful for the reader. In Chapter 2 we introduce the FETI-DP and BDDC method and also sketch the theory for the 2D scalar case to motivate our work on 3D linear elasticity in Chapter 3. There, we describe a FETI-DP algorithm for 3D linear elasticity along with an efficient parallel implementation, for which the theory was provided in Klawonn and Widlund [84]. We show the numerical and parallel scalability of the algorithm for academic benchmark problems as well as for more general situations with unstructured meshes from engineering problems where the theory does not apply. In Chapter 4

we introduce new, inexact FETI-DP algorithms. Here, different subproblems within the FETI-DP method can be solved approximately. Of particular interest is a version denoted irFETI-DP which allows for an inexact solution of the FETI-DP coarse problem. Using this approach the parallel scalability of FETI-DP algorithms can be extended further. We provide sequential and parallel experiments to demonstrate the high potential of the method and its applicability to problems with more than 10^4 subdomains. In Chapter 5 we describe a FETI-DP algorithm for elasticity with jumps in the material coefficient, also motivated by the theory in [84]. We investigate the effect of different material distributions guided by the theory and try to develop rules for more general cases than covered by the theory. In Chapter 6 we treat the case of two dimensional almost incompressible elasticity, and in Chapter 7 we apply our FETI-DP methods to spectral and *hp* finite element discretizations. We also show numerical and parallel scalability for the FETI-DP and irFETI-DP methods applied to spectral element discretizations.

The present chapter contains background material which is frequently needed in the presentation and analysis of the FETI method in subsequent chapters.

1.1 Spaces

Let $\Omega \subset \mathbf{R}^d$, $d = 2, 3$ be a bounded Lipschitz domain. We will be particularly interested in polygonal and polyhedral domains. Throughout this work we will refer to the diameter of a subdomain by H.

Define $L_2(\Omega)$ as the usual space of square integrable functions with the norm

$$\|u\|^2_{L_2(\Omega)} = \int_\Omega u^2 \, dx.$$

We have the standard Sobolev space $H^1(\Omega)$ as the space of functions which are square integrable as well as their first derivatives in the sense of distributions. Furthermore, we have the $H^1(\Omega)$-seminorm as

$$|u|^2_{H^1(\Omega)} = \int_\Omega \langle \nabla u, \nabla u \rangle \, dx.$$

We obtain a weighted norm suitable for our purposes by

$$\|u\|^2_{H^1(\Omega)} = |u|^2_{H^1(\Omega)} + H^{-2}\|u\|^2_{L_2(\Omega)},$$

where we introduce the weight factor to avoid undesirable scaling effects. We mention only briefly that the norms come from the corresponding inner

products.

We want to set essential boundary conditions on elements of $H^1(\Omega)$. We denote $C_0^\infty(\Omega)$ the infinitely differentiable functions with support in Ω. As the closure of $C_0^\infty(\Omega)$ in $\|\cdot\|_{H^1(\Omega)}$ we obtain the space $H_0^1(\Omega)$.

We recall that Sobolev spaces on manifolds can be introduced using sufficiently smooth maps and a partition of unity. Most notably, we will need the trace of $H^1(\Omega)$ on the domain boundary. The intrinsic $H^{1/2}(\Gamma)$-seminorm for $\Gamma \subset \partial\Omega$ is given by

$$|u|^2_{H^{1/2}(\Gamma)} = \int_\Gamma \int_\Gamma \frac{|u(x)-u(y)|^2}{|x-y|^d} d\Gamma(x) d\Gamma(y)$$

and the properly weighted norm on Γ by

$$\|u\|^2_{H^{1/2}(\Gamma)} = H^{-1}\|u\|^2_{L_2(\Gamma)} + |u|^2_{H^{1/2}(\Gamma)}.$$

Then, $H^{1/2}(\partial\Omega)$ is the trace space of $H^1(\Omega)$ on the domain boundary.

Note that an equivalent seminorm is given by

$$|u|^2_{H^{1/2}(\Gamma)} = \min_{v|\Gamma = u} |v|^2_{H^1(\Omega)} \tag{1.1}$$

as a result of the following theorems.

Theorem 1.1.1 (Trace Theorem) *If Ω is a bounded Lipschitz domain, then the trace operator $\gamma : u \to \gamma u$, which represents the restriction of u to the boundary of Ω, is a continuous mapping from $H^1(\Omega)$ onto $H^{1/2}(\partial\Omega)$. This implies*

$$\|\gamma u\|^2_{H^{1/2}(\partial\Omega)} \leq C(\Omega)\|u\|^2_{H^1(\Omega)}.$$

For a reference, see, e.g., [58, Theorem 1.5.1.3, p. 38]. This inequality remains valid if we substitute the norm by the seminorm.

Note that, due to the chosen scaling, we obtain

$$\|\gamma u\|^2_{L_2(\partial\Omega)} \leq C(\Omega)\left(H^{-1}\|u\|^2_{L_2(\Omega)} + H|u|^2_{H^1(\Omega)}\right).$$

We also have a family of extension theorems.

Theorem 1.1.2 (Extension Theorem) *There is a continuous extension operator from $H^{1/2}(\partial\Omega)$ to $H^1(\Omega)$, $u \mapsto \hat{u}$, therefore*

$$\|\hat{u}\|_{H^1(\Omega)} \leq C(\Omega)\|u\|_{H^{1/2}(\partial\Omega)}.$$

The harmonic extension, see Section 1.6, defines a continuous mapping in this sense. Sometimes we will identify functions from $H^{1/2}(\partial\Omega)$ with their harmonic extension to the interior.

The following well known inequalities, which establish the equivalence of norms in subspaces of $H^1(\Omega)$, are essential to us, see [102].

Theorem 1.1.3 (Friedrichs' Inequality) *There exists a constant $C(\Omega) > 0$ such that*

$$\|u\|^2_{H^1(\Omega)} \leq C(\Omega)|u|^2_{H^1(\Omega)}$$

for all $u \in H^1(\Omega, \Gamma) = \{u \in H^1(\Omega) : u = 0 \text{ on } \Gamma\}$, where $\Gamma \subset \partial\Omega$ is a surface with positive measure.

Theorem 1.1.4 (Poincaré Inequality) *There exists a constant $C(\Omega) > 0$ such that*

$$\|u\|^2_{H^1(\Omega)} \leq C(\Omega)\left(|u|^2_{H^1(\Omega)} + \frac{1}{H^{2+d}}\left(\int_\Omega u\,dx\right)^2\right)$$

for all $u \in H^1(\Omega)$. There are also Poincaré inequalities in $H^{1/2}(\Omega)$.

An additional space is relevant to this work since we want to consider isolated faces or edges of polyhedral domains.

Let Γ be an open subset of $\partial\Omega$ and $u \in H^{1/2}(\Gamma)$. Then the extension of u by zero from Γ to $\partial\Omega$ does not define a continuous operator from $H^{1/2}(\Gamma)$ to $H^{1/2}(\partial\Omega)$, see [58, pp. 18f, Lemma 1.3.2.6].

Therefore, we consider the space $H^{1/2}_{00}(\Gamma)$, defined as the set of all functions $u \in H^{1/2}(\Gamma)$ for which the continuation of u from Γ to $\partial\Omega$ with zero, denoted by $\tilde{u}$, remains bounded in $\|\cdot\|_{H^{1/2}(\partial\Omega)}$. Thus, we have

$$H^{1/2}_{00}(\Gamma) = \{u \in L_2(\Gamma) | \tilde{u} \in H^{1/2}(\partial\Omega)\}.$$

The space $H^{1/2}_{00}(\Gamma)$ is a subspace of $H^{1/2}_0(\Gamma)$ and complete with the norm inherited from $H^{1/2}(\partial\Omega)$ [58, pp. 18f, Lemma 1.3.2.6]. For our finite element functions, which are continuous by construction, this subtle difference disappears and we can either write $\|u\|^2_{H^{1/2}_{00}(\Gamma)}$ or $\|u\|^2_{H^{1/2}(\partial\Omega)}$.

Another equivalent norm for $H^{1/2}_{00}(\Gamma)$ is given by

$$\|u\|^2_{H^{1/2}_{00}(\Gamma)} = \|u\|^2_{H^{1/2}(\Gamma)} + \int_\Gamma u^2\, d(x;\Gamma)^{-1}\,dx,$$

where $d(x;\Gamma)$ denotes the distance from x to the boundary of Γ, see [58, pp. 18f].

1.2 Scalar Partial Differential Equations (PDEs)

1.2.1 Weak Formulation and Ellipticity

We obtain the weak formulation of a scalar partial differential equation of the type

$$\sum_{i,j} \partial_i(a_{ij}(x)\,\partial_j u) = f \quad \text{in } \Omega, \tag{1.2}$$

$$u = 0 \quad \text{on } \partial\Omega_D \subset \partial\Omega, \tag{1.3}$$

and

$$\sum_{i,j} a_{ij}(x)\,\partial_j u\,\nu_i = g \quad \text{on } \Gamma_N = \partial\Omega \setminus \partial\Omega_D, \tag{1.4}$$

where ν is the outward normal to $\partial\Omega$, by means of a Green's formula. We get

$$\int_\Omega \sum_{i,j} a_{ij}(x)\,\partial_i v \partial_j u\,dx = \int_\Omega fv\,dx + \int_{\partial\Omega_N} gv\,ds, \tag{1.5}$$

for all $v \in H_0^1(\Omega, \partial\Omega_D) := \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega_D\}$. The differentiation ∂ is to be understood in the variational sense.

We introduce the symmetric bilinear form $a(u, v)$, the linear form $f(v)$, and the Hilbert space

$$V := H_0^1(\Omega, \partial\Omega_D)$$

to write (1.5) in the compact form

$$a(u, v) = f(v) \quad \forall v \in V. \tag{1.6}$$

The Lemma of Lax-Milgram guarantees that the problem (1.6) has a unique solution if $a(\cdot, \cdot)$ is continuous,

$$|a(u, v)| \le C\|u\|\|v\| \quad \forall u, v \in V,$$

elliptic

$$\exists \alpha > 0 : a(u, u) \ge \alpha\|u\|^2 \quad \forall u \in V,$$

and $f(\cdot)$ is continuous,

$$|f(v)| \le C\|v\| \quad \forall v \in V.$$

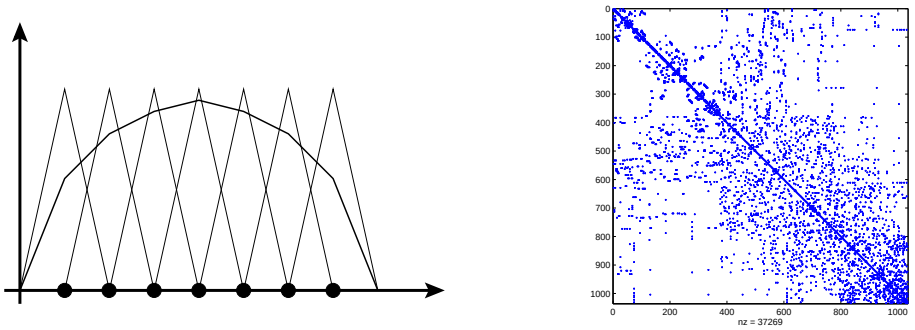

Figure 1.1
Linear basis functions and a linear combination in one dimension (left). Sparsity pattern of a stiffness matrix for a 3D problem (right).

1.2.2 Finite Element Method (FEM)

Consider the weak formulation of a homogeneous second order elliptic problem, e.g. the scalar equation

$$\int_\Omega \sum_{i,j} a_{ij}(x)\, \partial_i u \partial_j v \, dx = \int_\Omega f v \, dx \quad \forall v \in V, \tag{1.7}$$

or in compact notation

$$a(u, v) = (f, v) \quad \forall v \in V. \tag{1.8}$$

In the finite element method we replace the infinite dimensional function space V by a finite dimensional subspace V^h, e.g. the space of piecewise linear continuous functions. This yields

$$a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V^h. \tag{1.9}$$

The basis representation $u_h = \sum_i u_i \phi_i$ leads to the linear equation system

$$Ku = b,$$

where $K = \Big(a(\phi_i, \phi_j) \Big)$ and $b_i = \Big((f, \phi_i) \Big)$.

The matrix is symmetric positive definite if the bilinear form $a(\cdot, \cdot)$ is symmetric and V-elliptic, and it is sparse since the basis functions only have a local support. The number of nonzero entries is generally $O(n)$ if the size of K is $n \times n$.

1.3 Direct Solvers

The simplest algorithm to solve a system of linear equations

$$Ax = b,$$

where $A \in \mathbf{R}^{m\times m}$, is the well known Gaussian elimination,

```
for i=1..m
   for j=i+1..m
      for k=i+1..m
         a(j,k)=a(j,k)-a(i,k)*a(j,i)/a(i,i)
      end
   end
end,
```

which requires $O(m^3)$ operations. Here, we have given the Gaussian elimination in its simplest, so called right-looking, form. Finite element stiffness matrices are typically sparse since the basis functions of the finite element space are local. Hence, it is more efficient to store only nonzeros entries in the matrix. The Gaussian elimination then takes the form

```
for i=1=1..n
   for j=nextnonzeroindex(j)
      for k=nextnonzeroindex(k)
         a(j,k)=a(j,k)-a(i,k)*a(j,i)/a(i,i)
      end
   end
end.
```

Here, the computational cost has been significantly reduced but during the elimination process some entries in A which were previously zero become nonzero. These new entries are referred to as *fill-in*. The amount of fill-in in the elimination process depends on the ordering of the matrix and increases the computational cost.

Let us assume that we can order our matrix in a way that it has the following block structure

$$\begin{bmatrix} K_{II}^{(1)} & & & \widetilde{K}_{I\Gamma}^{(1)} \\ & \ddots & & \vdots \\ & & K_{II}^{(N)} & \widetilde{K}_{I\Gamma}^{(N)} \\ \widetilde{K}_{\Gamma I}^{(1)} & \cdots & \widetilde{K}_{\Gamma I}^{(N)} & \widetilde{K}_{\Gamma\Gamma} \end{bmatrix} = \begin{bmatrix} K_{II} & \widetilde{K}_{I\Gamma} \\ \widetilde{K}_{\Gamma I} & \widetilde{K}_{\Gamma\Gamma} \end{bmatrix}.$$

We have chosen this notation because we can think of the matrices $K_{II}^{(i)}$ as subdomain interior matrices and of $\widetilde{K}_{\Gamma\Gamma}$ as the global interface matrix. All

the matrices $K_{II}^{(i)}$ can be factored independently (possibly also in parallel), and no fill-in will appear in the off-diagonal blocks of

$$\begin{bmatrix} K_{II}^{(1)} & & \\ & \ddots & \\ & & K_{II}^{(N)} \end{bmatrix}.$$

Then, finally, the Schur complement $\mathcal{S} = \widetilde{K}_{\Gamma\Gamma} - \widetilde{K}_{\Gamma I} K_{II}^{-1} \widetilde{K}_{I\Gamma}$ has to be formed and the Schur complement system has to be solved. The matrix $\mathcal{S}$ contains many dense blocks and we could perform this last step even by using dense Gaussian elimination, see above. Moreover, we can apply this approach recursively to $K_{II}^{(i)}$ until our subdomains become single matrix entries.

For this approach to be efficient, the matrix $\widetilde{K}_{\Gamma\Gamma}$ should be small. Unfortunately, the problem to find the optimal interface, also called separator, is NP complete. Therefore, different strategies have been developed to find good orderings of K, e.g. nested dissection (ND) [50] and minimum degree (MD/AMD) [51, 27] orderings and many variants thereof. In finding good separators, direct solvers have also benefitted, as have iterative substructuring methods, from advances in graph partitioning software.

Applying optimal orderings, the complexity of directly solving linear systems resulting from finite element discretizations can be reduced to $O(m^{3/2})$ in 2D and $O(m^2)$ in 3D.

The design and the implementation of efficient direct solvers is a challenging task, and significant progress has been made in recent years. Today several efficient direct solver packages are available. Among them are MUMPS [5] and UMFPACK [27], which both use the multifrontal approach [33]. The package UMFPACK is a sequential code whereas MUMPS is also targeted at distributed memory parallel computers. We have also used Spooles [7] and SuperLU_Dist [93]. The latter is the distributed memory version of the supernodal direct solver package SuperLU. Other (sequential and parallel) direct solvers, not available in source code, are WSMP [64] and PARDISO [115].

1.4 Krylov Subspace Methods

Domain decomposition methods are commonly used as preconditioners for the use with Krylov subspace methods. We will briefly describe the two methods which are of most relevance to this work. For further details, refer to Saad [113], Greenbaum [57], and many others. It is well known that

for well conditioned problems the performance of different Krylov subspace methods is very similar.

1.4.1 Preconditioned Conjugate Gradient Method (PCG)

Let us consider a symmetric positive definite linear system

$$Ax = b. \tag{1.10}$$

The Preconditioned Conjugate Gradient (PCG) method is an iterative method to solve (1.10) by minimizing

$$\frac{1}{2}x^T M^{-1/2} A M^{-1/2} x - x^T M^{-1/2} b,$$

over a certain Krylov subspace; see, e.g. [113, 57, 123]. Here, the symmetric operator M^{-1} is the preconditioner for A.

The convergence of the PCG method is governed by the eigenvalues of $M^{-1}A$, i.e.

$$\frac{\|x^{(k)} - x^*\|_A}{\|x^{(0)} - x^*\|_A} \leq 2 \left(\frac{\sqrt{\kappa(M^{-1}A)} - 1}{\sqrt{\kappa(M^{-1}A)} + 1} \right)^k,$$

where $\kappa(M^{-1}A) = \frac{\lambda_{\max}(M^{-1}A)}{\lambda_{\min}(M^{-1}A)}$ is the condition number of $M^{-1}A$.

Both, the system matrix A as well as the preconditioning matrix M^{-1}, are only needed in matrix vector multiplications. Therefore, $M^{-1}x$ as well as Ax must be easy to compute, and the condition number $\kappa(M^{-1}A)$ should be small.

If elliptic partial differential equations of second order are discretized with the finite element method, the system matrix is generally not well conditioned, i.e. the condition number of A is proportional to $1/h^2$, where h is a measure of the element diameter. In higher order methods the system matrices can be even more ill conditioned, and the condition number depends on the polynomial degree. However, for an efficient algorithm the condition number of $M^{-1}A$ should be scalable with respect to the mesh size and, possibly, also with respect to the spectral degree of the discretization. This is the case for the iterative substructuring methods discussed in this work and also for other domain decomposition methods and subspace correction schemes.

In this work, whenever conjugate gradients are mentioned, we actually refer to the Preconditioned Conjugate Gradient method.

1.4.2 Generalized Minimal Residual Method (GMRES)

The Generalized Minimal Residual (GMRES) method [114] is an iterative method to solve an unsymmetric linear system

$$Ax = b$$

by minimizing the residual over a certain Krylov subspace. If a preconditioner M^{-1} is available, either left of right preconditioning can be used with GMRES. As opposed to conjugate gradients, where a three term recurrence is used, the work and storage grows linearly with the number of iterations. A full basis for the Krylov subspace has to be computed and stored. Therefore, often a truncated version (restarted GMRES) is applied.

Unlike in CG, the convergence behavior of GMRES cannot be described by the eigenvalues alone. If the field of values

$$F(A) := \left\{ \frac{y^T M^{-1} A y}{y^T y} : y \in \mathbf{R}^n \right\}$$

is contained in a disk in the complex plane of diameter s which is centered at c then, see [57],

$$\frac{\|r_k\|}{\|r_0\|} \leq 2 \left(\frac{s}{|c|} \right)^k .$$

For symmetric preconditioners M^{-1} in [128] an estimate similar to a classical bound by Elman [34] has been given,

$$\frac{\|r_k\|}{\|r_0\|} \leq \Big(1 - F\big(M^{-1}A\big) F\big((M^{-1}A)^{-1}\big) \Big)^{k/2} .$$

For block triangular preconditioners in [80] a field of value analysis has been given, using equivalent norms.

1.5 Scalability

We are interested in solving linear equation systems obtained from the discretization of partial differential equations. We consider an algorithm *scalable* if the computational work to find a solution of a prescribed accuracy, e.g. an accuracy of 10^{-7}, grows linearly with the size of the linear system, i.e. the number of degrees of freedom (d.o.f.).

In domain decomposition methods accelerated by a Krylov subspace methods like CG or GMRES, if we neglect the computational work for the small coarse problem, we have scalability in this sense if the number of iterations

is independent of the problem size. For CG this is the case if the condition number is bounded independently of the number of subdomains.

Additionally, we require for our domain decomposition methods that the number of iterations only depends weakly, i.e. polylogarithmically, on the diameter of the subdomains. This gives us some freedom to choose the subdomain sizes. If a domain decomposition method fulfills these two conditions, we call it *numerically scalable*.

We also consider two types of parallel scalability. In the first, we consider a problem of fixed size and increase the number of processors used to solve the problem. An algorithm is *parallel scalable* if, ideally, twice as many processors will solve the problem in half the time. In the second type of scalability, which is sometimes called weak scalability, we double the problem size and the number of processors at the same time. We then require that, in a parallel scalable algorithm, the solution time is independent of the problem size.

The first type of scalability is generally more difficult to achieve in real world applications. The second type of scalability has become important with the advent of massively parallel computers. It explains why supercomputers with thousands of processors can still be used efficiently.

1.6 Discrete Harmonic Functions

Let us recall that in the same way as the harmonic functions in $H^1(\Omega)$ are identified through the orthogonality to $H_0^1(\Omega)$,

$$(\nabla u, \nabla v) = 0 \quad \forall v \in H_0^1(\Omega), \tag{1.11}$$

the discrete harmonic functions are identified through

$$(\nabla u_h, \nabla v_h) = 0 \quad \forall v_h \in V_0^h(\Omega), \tag{1.12}$$

or explicitly $K_{II}u_I + K_{I\Gamma}u_\Gamma = 0$. This expresses the orthogonality of the functions associated with the the boundary values and the interior functions in the norm defined by the symmetric positive definite stiffness matrix

$$K = \begin{pmatrix} K_{II} & K_{I\Gamma} \\ K_{\Gamma I} & K_{\Gamma\Gamma} \end{pmatrix}.$$

The subscript Γ refers to boundary variables and the subscript I to interior variables.

It follows from the orthogonality that harmonic and discrete harmonic functions are the extensions of their boundary values with minimal energy in

their respective spaces. For the harmonic function u and discrete harmonic function u_h, we have

$$|u|_{H^1(\Omega)} = \inf_{v-u \in H_0^1(\Omega)} |v|_{H^1(\Omega)} \tag{1.13}$$

and

$$|u_h|_{H^1(\Omega)} = \inf_{v_h - u_h \in V_0^h(\Omega)} |v_h|_{H^1(\Omega)}, \tag{1.14}$$

since for $u - v \in H_0^1(\Omega)$

$$(\nabla v, \nabla v) = (\nabla(v - u + u), \nabla(v - u + u)) \geq (\nabla u, \nabla u) \tag{1.15}$$

is true as well as the analogous result for u_h.

We denote the discrete harmonic extension of a function $u \in H^{1/2}(\partial\Omega)$ to the interior by $\mathcal{H}u \in H^1(\Omega)$.

1.7 Equivalence of Schur Complement Norm and $H^{1/2}(\partial\Omega)$-Norm

Assuming that Ω is a Lipschitz domain, the seminorm defined by the Schur complement is equivalent to the properly scaled $H^{1/2}(\partial\Omega)$-seminorm. See [16], the extension in [143], and see also [136].

Considering the bilinear form

$$\int_\Omega \rho \langle \nabla u, \nabla v \rangle \, dx,$$

related to the Poisson equation with a constant coefficient ρ, we obtain

$$\frac{1}{C}\rho |u_h|^2_{H^{1/2}(\partial\Omega)} \leq |u_h|^2_S \leq C\rho |u_h|^2_{H^{1/2}(\partial\Omega)}. \tag{1.16}$$

1.8 Graphs

A graph or undirected graph $\mathfrak{G}$ is an ordered pair $\mathfrak{G} = (V, E)$, where $V \subset \mathbf{N}$ is the set of nodes, and E is a set of unordered pairs $\{v_1, v_2\}$, where $v_1, v_2 \in V$ are nodes. The elements of $e = \{v_1, v_2\}$ of E are called edges.

We say that nodes $v_1 \in V$ and $v_2 \in V$ are connected through the edge e if $e = \{v_1, v_2\} \in E$.

A path from $v_1 \in V$ to $v_2 \in V$ is a list of nodes, starting with v_1 and ending with v_2, where all pairs of consecutive nodes are connected through

edges. If a path from $v_1 \in V$ to $v_2 \in V$ exists, we say that v_1 and v_2 are connected in $\mathfrak{G}$. Connectivity is an equivalence relation over V and the equivalence classes are called connectivity components of $\mathfrak{G}$.

Efficient (sequential) algorithms exist to compute connectivity components of graphs, i.e. depth first search or breadth first search, see any introduction into efficient algorithms, e.g., [120], and also [2].

A simple path in $\mathfrak{G}$ is a path in $\mathfrak{G}$, where no node $v \in V$ is repeated. A cycle in $\mathfrak{G}$ is a simple path, where start and end node coincide. We call a graph without cycles a forest and a connected forest a tree.

A subgraph $\tilde{\mathfrak{G}} = (\tilde{V}, \tilde{E})$ of $\mathfrak{G} = (V, E)$ is a graph where $\tilde{V} \subset V$ and $\tilde{E} \subset E$. We say the subgraph $\tilde{\mathfrak{G}}$ spans $\mathfrak{G}$ if $\tilde{V} = V$. A spanning tree of $\mathfrak{G} = (V, E)$ is a subgraph of $\mathfrak{G} = (V, E)$ which spans $\mathfrak{G}$ and is a tree.

A minimal spanning tree is a spanning tree with the smallest possible number of edges. Sometimes a cost function can be associated with edges, in this case the spanning tree minimizes the sum of the edge costs. Efficient algorithms are known to calculate minimal spanning trees.

A graph is planar if it can be drawn in a plane with no crossing edges.

Chapter 2

Two Iterative Substructuring Methods

In this chapter, we describe the two well known iterative substructuring methods relevant for this work. For simplicity, we will introduce the algorithms in the context of 2D scalar elliptic problems, and we will also sketch the convergence theory in this context based on publications of several other authors [100, 84]. This will serve as motivation for our main work on 3D elasticity problems in Chapter 3 and 5.

Dual-primal FETI (FETI-DP) methods are the most recent members of the family of Finite Element Tearing and Interconnecting (FETI) domain decomposition methods. The FETI methods, introduced by Farhat and Roux [45], are dual iterative substructuring methods for partial differential equations. In these methods the original domain, on which the given partial differential equation has to be solved, is decomposed into nonoverlapping subdomains. The intersubdomain continuity is then enforced by Lagrange multipliers across the interface defined by the subdomain boundaries. For further results and references, see, e.g., [46, 38, 13, 99, 83, 136].

In dual-primal FETI methods, some continuity constraints on the primal displacement variables are forced to hold throughout iterations, as in primal substructuring algorithms, while the other constraints are enforced by the use of Lagrange multipliers, as in FETI. The primal constraints have to be chosen such that the local subproblems become invertible and such that a parallel scalable method is obtained. The primal constraints provide a coarse problem for these domain decomposition methods.

Dual-primal FETI (FETI-DP) algorithms were introduced by Farhat et al. in [40] for linear elasticity problems in the plane and then extended by Farhat, Lesoinne, and Pierson [41] to three dimensional elasticity problems; see also Pierson [107]. The first theoretical analysis for two dimensional,

scalar elliptic partial differential equations of second and fourth order with only small coefficient jumps across the subdomain boundaries was given by Mandel and Tezaur [100]. It was shown that the condition number is bounded polylogarithmically as a function of the dimension of the individual subregion problems. The family of algorithms for scalar, second order elliptic problems in three dimensions was extended by Klawonn, Widlund, and Dryja [82, 86, 87]; see also [136]. There, constraints on averages over subdomain edges and faces are used to obtain algorithms with an improved scalability for three dimensional problems. In [41, 107] similar constraints on averages are used for linear elasticity problems, although the definition of subdomain edges is different from the one used in [82, 86, 87], see also the discussion in Section 3.3.3. We also note that an important feature of the algorithms presented in [82, 86, 87] was to develop variants of FETI-DP with smaller coarse problems while still maintaining scalability and a good convergence bound. In fact, in [86, 87] a theory was provided which shows that the condition number in three dimensions can again be bounded polylogarithmically as a function of the dimension of the individual subdomain problems and that the bounds can otherwise be made independent of the number of subdomains, the mesh size, and the jumps in the coefficients.

More recently, new variants of FETI-DP for three dimensional, linear elasticity problems were provided by Klawonn and Widlund [84, 85] together with a theoretical analysis proving a polylogarithmic condition number estimate as in the scalar case which is also robust with respect to discontinuities in the material coefficients. For benign elasticity problems, it is shown that selecting an appropriate set of edge averages as primal constraints is sufficient to obtain good polylogarithmic bounds. For arbitrary coefficient distributions, certain first order moments on selected edges have to be added as primal constraints as well as constraints at some of the vertices in order to obtain robust, polylogarithmic bounds.

Strongly related to FETI-DP methods are the more recently developed Neumann-Neumann methods with constraints, also known as the Balancing Domain Decomposition methods by Constraints (BDDC); cf. [29, 97, 98]. Neumann-Neumann methods with primal vertex constraints were also developed independently by Cros [26]. It was first proved in Mandel, Dohrmann, and Tezaur [98] that the eigenvalues of BDDC and FETI-DP, which are not one or zero, are the same; see also Li and Widlund [92] and Brenner and Sung [20] for other approaches.

Recently, new results for inexact FETI-DP and BDDC algorithm have been obtained for the inexact solution of the subdomain or coarse problems. For inexact BDDC methods, see Tu [138, 139, 140], Dohrmann [31], and Li and Widlund [91], and for inexact FETI-DP algorithms, see Chapter 4.

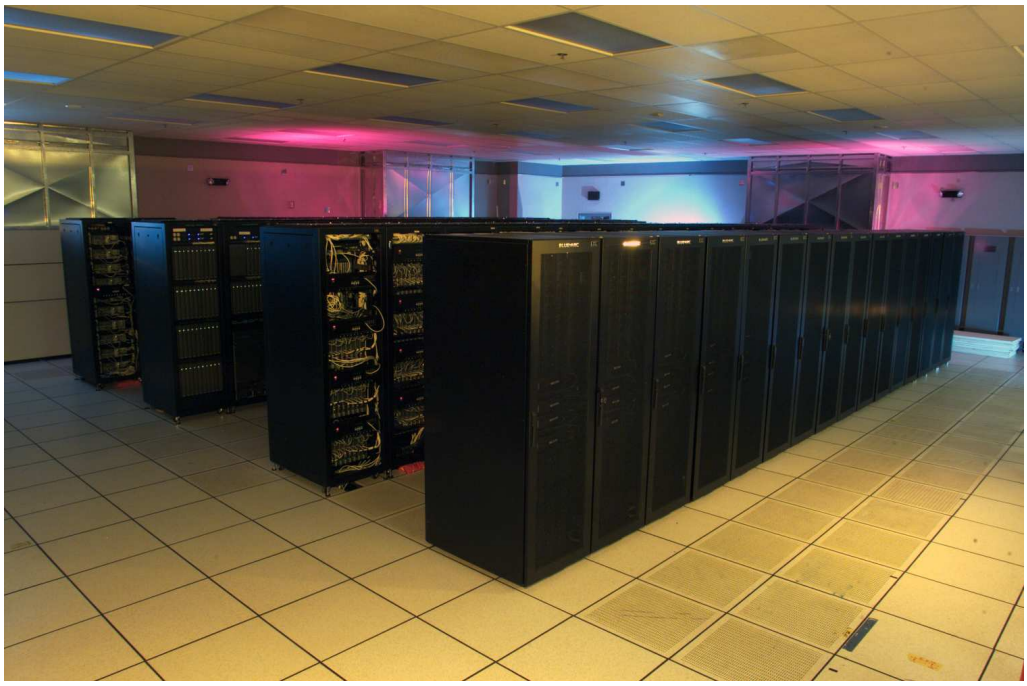

Figure 2.1
MCR supercomputer at Lawrence Livermore National Laboratory (Image: Lawrence Livermore National Laboratory, California, USA).

2.1 A First Model Problem

We will first use a simple model problem in order to define the setting for our presentation of the FETI-DP method and the Neumann-Neumann method with constraints, also known as the BDDC method. Later we will consider the equations of linear elasticity. By using a change of basis, if necessary, the presentation of the algorithms at the level of linear algebra can be the same for all second order elliptic PDEs, but the theory is significantly easier to present for scalar equations. Note that the choice of the primal constraints depends in the spatial dimension and the problem type.

Our presentation will be biased towards the FETI-DP method. Since Mandel and Dohrmann [98] have shown that FETI-DP and BDDC share all but possibly two of their eigenvalues, this is not a limitation. See also Li and Widlund for a proof which is closer to our notation. Already in Klawonn and Widlund [83] a common theory for methods of the older FETI and Neumann-Neumann type has been given.

We consider the following second order scalar elliptic problem in a two dimensional region $\Omega \subset \mathbf{R}^2$: Find $u \in H_0^1(\Omega, \partial\Omega_D)$ such that

$$\int_\Omega \rho(x)\nabla u \cdot \nabla v \, dx = \int_\Omega fv \quad \forall v \in H_0^1(\Omega, \partial\Omega_D).$$

We decompose Ω into N nonoverlapping subdomains $\Omega_i, i = 1, \ldots, N$, i.e.

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i \,, \qquad \Omega_i \cap \Omega_j = \emptyset \quad \text{if} \quad i \neq j \,.$$

Each subdomain is the union of shape-regular finite elements with matching finite element nodes across the interface,

$$\Gamma := \bigcup_{i \neq j} \partial\Omega_i \cap \partial\Omega_j,$$

where $\partial\Omega_i, \partial\Omega_j$ are the boundaries of Ω_i, Ω_j, respectively. These subdomains are also often referred to as substructures. The coefficient $\rho(x) > 0$ can be discontinuous but should have large coefficient jumps only across the interface. In the standard theory it is assumed that $\rho(x)$ is constant on each subdomain. Later, we will see in our numerical simulations that this assumption is in fact too strong, and in many cases large coefficient jumps within subdomains can be allowed without affecting the convergence of the method at all.

We assume that subdomains are shape regular and have a typical diameter of H whereas the finite elements are of typical diameter h. We will use low

order conforming finite elements, i.e. linear or bilinear finite elements. In Chapter 7 we will also consider higher order, *hp* or spectral, elements.

We define the bilinear form

$$a(u,v) := \int_\Omega \rho(x)\nabla u \cdot \nabla v \, dx$$

and the functional

$$f(v) := \int_\Omega f v \, dx.$$

We denote by $W_g^h := W_g^h(\Omega)$ the conforming finite element space of finite element functions. The associated discrete problem is then

$$a(u_h, v_h) = f(v) \quad \forall v_h \in W_g^h. \tag{2.1}$$

When there is no risk of confusion, we will drop the subscript h. Moreover, we will often identify the finite element function u_h and the corresponding vector $u = (u_1, \ldots, u_n)^T \in \mathbf{R}^n$, where $u_h = \sum_{i=1}^N u_i \phi_i$, and where $\phi_1, \ldots, \phi_N$ is the finite element basis we choose to work in.

For each subdomain Ω_i we will denote the finite element space by W_i. In iterative substructuring algorithms the interior variables in each subdomain are eliminated using a direct solver. It is therefore helpful to partition W_i into the interior part $W_I^{(i)}$ and the finite element trace space $W_\Gamma^{(i)}$, i.e. we have

$$W_i = W_I^{(i)} \oplus W_\Gamma^{(i)}.$$

Note that we consider variables on the Neumann boundary $\partial\Omega_N$ as interior to a subdomain. Also, subdomains not intersecting $\partial\Omega_D$ lack essential boundary conditions and will have singular stiffness matrices. We introduce the product spaces

$$W := \prod_{i=1}^N W_i$$

and

$$W_\Gamma := \prod_{i=1}^N W_\Gamma^{(i)}.$$

The members of W and W_Γ are in general not continuous across the interface. However, we expect our global solution to be continuous. We therefore consider the subspace formed by the continuous functions in W, which we denote by $\widehat{W}$, i.e. we have

$$\widehat{W} := \{w \in W : w \text{ is continuous across } \Gamma\}.$$

We also define $\widehat{W}_\Gamma$ as the subspace formed by the continuous members of W_Γ, i.e.

$$\widehat{W}_\Gamma := \{w_\Gamma \in W_\Gamma : w_\Gamma \text{ is continuous across } \Gamma\}.$$

In our iterative substructuring methods, we will iterate in the space W while requiring certain constraints to hold throughout the iteration. In FETI-DP methods these constraints are continuity constraints, also called primal constraints, that have to be chosen such that each subdomain problem becomes invertible and that also a good convergence bound can be obtained. We denote the space of finite element functions which fulfill these constraints by $\widetilde{W}$. We have

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

and likewise

$$\widehat{W}_\Gamma \subset \widetilde{W}_\Gamma \subset W_\Gamma.$$

The continuity constraints posed on $\widetilde{W}$ only affect a subset of the variables, these variables are denoted as *primal*. We can decompose W_Γ into the primal space $W_\Pi = \prod_{i=1}^N W_\Pi^{(i)}$ and the dual space $W_\Delta = \prod_{i=1}^N W_\Delta^{(i)}$, i.e.

$$W_\Gamma = W_\Pi \oplus W_\Delta.$$

Using this notation, we can now decompose $\widetilde{W}_\Gamma$ into the subspace $\widehat{W}_\Pi$ which has continuous members only and the subspace W_Δ which also contains finite element functions which are not continuous, i.e. we have

$$\widetilde{W}_\Gamma = \widehat{W}_\Pi \oplus W_\Delta.$$

Likewise, we can now write

$$\widehat{W} = \widehat{W}_\Pi \oplus \widehat{W}_\Delta \oplus W_I.$$

Here, $\widehat{W}_\Delta$ is the subspace formed by the continuous finite element functions in W_Δ. For convenience, we also introduce the nonprimal space

$$W_B = W_\Delta \oplus W_I$$

and

$$\widehat{W}_B = \widehat{W}_\Delta \oplus W_I.$$

2.2 FETI-DP and BDDC with Matrices and Vectors

We will now introduce a matrix formulation of the FETI-DP and the BDDC method.

As we consider a domain $\Omega \subset \mathbf{R}^2$, the interface Γ is the union of edges and vertices. In the continuous setting, we regard edges in 2D as open sets that are shared by two subdomains, and vertices as endpoints of edges, see, e.g. Toselli and Widlund [136, Chapter 4.2]. In Chapter 3 we give a detailed definition of vertices, edges, and faces in 3D which is also suitable for irregular domain decompositions.

For each subdomain $\Omega_i,\ i = 1, \ldots, N$, we assemble the local stiffness matrices $K^{(i)}$ and load vectors $f^{(i)}$. We denote the unknowns on each subdomain by $u^{(i)}$.

We then partition the unknowns $u^{(i)}$ into interface variables $u_\Gamma^{(i)}$, i.e. unknowns which are associated with the interface Γ, and interior unknowns $u_I^{(i)}$, which are interior to the subdomain Ω_i. Note that any variable on $\partial\Omega_N \setminus \Gamma$ is considered to be interior to a subdomain. Unknowns on $\partial\Omega_D$ are usually eliminated upfront and do not need to be considered.

We then partition the interface variables into primal variables $u_\Pi^{(i)}$ and dual variables $u_\Delta^{(i)}$. In two dimensional problems the primal variables $u_\Pi^{(i)}$ will be associated with vertex unknowns. Vertex unknowns are also often referred to as corner variables. In a structured decomposition of a unit square into smaller squares, the vertex unknowns are all degrees of freedom (d.o.f.) which are associated with nodes shared by four subdomains. Together, interior and dual unknowns form the nonprimal variables $u_B^{(i)}$.

In the FETI-DP algorithms we will enforce the continuity of the solution in the primal unknowns $u_\Pi^{(i)}$ by global subassembly of the subdomain stiffness matrices $K^{(i)}$. For all other interface variables $u_\Delta^{(i)}$ we will introduce Lagrange multipliers to enforce continuity.

We also partition the stiffness matrices according to the different sets of unknowns,

$$K^{(i)} = \begin{bmatrix} K_{BB}^{(i)} & K_{\Pi B}^{(i)T} \\ K_{\Pi B}^{(i)} & K_{\Pi\Pi}^{(i)} \end{bmatrix}, \quad f^{(i)} = \begin{bmatrix} f_B^{(i)} \\ f_\Pi^{(i)} \end{bmatrix},$$

and

$$K_{BB}^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} \end{bmatrix}, \quad f_B^{(i)} = \begin{bmatrix} f_I^{(i)} \\ f_\Delta^{(i)} \end{bmatrix}.$$

We will refer to the completely assembled global stiffness matrix as K_g.

2.2.1 FETI-DP Method

We define the block matrices

$$K_{BB} := \operatorname{diag}_{i=1}^{N}(K_{BB}^{(i)}),$$

$$K_{\Pi B} := \mathrm{diag}_{i=1}^{N}(K_{\Pi B}^{(i)}),$$

$$K_{\Pi\Pi} := \mathrm{diag}_{i=1}^{N}(K_{\Pi\Pi}^{(i)}),$$

and the right hand sides

$$f_B^T := [f_B^{(1)T}, \ldots, f_B^{(N)T}],$$

$$f_\Pi^T := [f_\Pi^{(1)T}, \ldots, f_\Pi^{(N)T}].$$

By assembly of the local subdomain matrices in the primal variables using the operator $R_\Pi^T = [R_\Pi^{(1)T}, \ldots, R_\Pi^{(N)T}]$ with entries 0 or 1, we have the partially assembled global stiffness matrix

$$\widetilde{K} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix} = \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi^T \end{bmatrix} \begin{bmatrix} K_{BB} & K_{\Pi B}^T \\ K_{\Pi B} & K_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi \end{bmatrix}$$

and right hand side

$$\tilde{f} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \end{bmatrix} = \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi^T \end{bmatrix} \begin{bmatrix} f_B \\ f_\Pi \end{bmatrix}.$$

Choosing a sufficient number of primal variables $u_\Pi^{(i)}$ to constrain our solution, for example all vertex unknowns, results in a symmetric positive definite matrix $\widetilde{K}$. We note that the upper left block of

$$\widetilde{K} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix}$$

is block diagonal and coupling is introduced only through the primal variables.

To enforce continuity on the remaining interface variables $u_\Delta^{(i)}$ we introduce a jump operator B_B with entries $0, -1$ or 1 and Lagrange multipliers λ.

We can now formulate the FETI-DP saddle-point problem

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B_B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}. \tag{2.2}$$

By eliminating u_B and u_Π from the system (2.2) we obtain the linear system

$$F\lambda = d,$$

where

$$F := \left[\begin{array}{cc} B_B & 0 \end{array} \right] \left[\begin{array}{cc} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{array} \right]^{-1} \left[\begin{array}{c} B_B^T \\ 0 \end{array} \right],$$

$$d := \left[\begin{array}{cc} B_B & 0 \end{array} \right] \left[\begin{array}{cc} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{array} \right]^{-1} \left[\begin{array}{c} f_B \\ \tilde{f}_\Pi \end{array} \right].$$

From

$$\left[\begin{array}{cc} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{array} \right]^{-1} = \left[\begin{array}{cc} I_B & -K_{BB}^{-1}\widetilde{K}_{\Pi B}^T \\ 0 & I_\Pi \end{array} \right] \left[\begin{array}{cc} K_{BB}^{-1} & 0 \\ 0 & \widetilde{S}_{\Pi\Pi}^{-1} \end{array} \right] \left[\begin{array}{cc} I_B & 0 \\ -\widetilde{K}_{\Pi B}K_{BB}^{-1} & I_\Pi \end{array} \right]$$

we conclude that

$$\begin{aligned} F &= B_B K_{BB}^{-1} B_B^T + B_B K_{BB}^{-1} \widetilde{K}_{B\Pi} \widetilde{S}_{\Pi\Pi}^{-1} \widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T, \\ d &= B_B K_{BB}^{-1} f_B + B_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \widetilde{S}_{\Pi\Pi}^{-1} (\tilde{f}_\Pi - \widetilde{K}_{\Pi B} K_{BB}^{-1} f_B). \end{aligned}$$

In the standard, exact FETI-DP methods two different preconditioners for F are commonly used, the theoretically almost optimal Dirichlet preconditioner M_{FETI_D} and the lumped preconditioner M_{FETI_L}.

We define additional block matrices

$$\begin{aligned} K_{II} &:= \mathrm{diag}_{i=1}^N(K_{II}^{(i)}), \\ K_{\Delta I} &:= \mathrm{diag}_{i=1}^N(K_{\Delta I}^{(i)}), \\ K_{\Delta\Delta} &:= \mathrm{diag}_{i=1}^N(K_{\Delta\Delta}^{(i)}). \end{aligned} \tag{2.3}$$

The Dirichlet preconditioner M_{FETI_D} is then defined by

$$M_{\mathrm{FETI}_D}^{-1} := B_{B,D}(R_\Delta^B)^T(K_{\Delta\Delta} - K_{\Delta I}K_{II}^{-1}K_{\Delta I}^T)R_\Delta^B B_{B,D}^T \tag{2.4}$$

and the lumped preconditioner M_{FETI_L} by

$$M_{\mathrm{FETI}_L}^{-1} = B_{B,D}(R_\Delta^B)^T K_{\Delta\Delta} R_\Delta^B B_{B,D}^T, \tag{2.5}$$

where

$$R_\Delta^B = \mathrm{diag}_{i=1}^N(R_\Delta^{B\,(i)}).$$

The matrices $R_\Delta^{B\,(i)}$ are restriction matrices with entries 0 or 1 which restrict the nonprimal degrees of freedom $u_B^{(i)}$ of a subdomain to the dual part $u_\Delta^{(i)}$. The matrices $B_{B,D}$ are scaled variants of the jump operator B_B where the contribution from and to each interface node is scaled by the inverse of the multiplicity of the node. The multiplicity of a node is defined as the number

of subdomains it belongs to. The scaling is chosen such that the linear mapping $P_{B,D} = B_{B,D}^T B_B$ becomes a projection. It is well known that for heterogeneous problems a more elaborate scaling is necessary, see Section 2.3.

The original or standard, exact FETI-DP method is the method of conjugate gradients applied to the symmetric positive definite system

$$F\lambda = d$$

with the preconditioners $M^{-1}_{\mathrm{FETI}_D}$ or $M^{-1}_{\mathrm{FETI}_L}$. Only the Dirichlet preconditioner has almost optimal theoretical properties, i.e. we have a polylogarithmic condition number bound, see Sections 2.4 and 3.3.5.

The symmetric positive definite Schur complement $\widetilde{S}_{\Pi\Pi}$ represents the FETI-DP coarse problem; it introduces the global coupling across the subdomains. It is essential for the numerical scalability of FETI-DP and we will see later that this matrix also represents the coarse problem of the BDDC method.

The inverse matrices that appear in the matrix description of the FETI-DP method are never calculated as they are only needed in terms of matrix vector multiplications in the Krylov subspace method. Instead, direct solvers, i.e. Cholesky or LU decompositions, are used.

The FETI-DP method is very well suited for parallel computing. A very large share of the work, i.e. the factorization of the block matrix K_{BB} as well as the corresponding forward backward substitutions, can be carried out independently for each subdomain and thus are completely parallel. Only a very small part of the work, i.e. the assembly and factorization of the coarse matrix $\widetilde{S}_{\Pi\Pi}$, is global and more challenging to parallelize. Additionally, we have communication only at the surface of subdomains, i.e. whenever the operators B, B^T are applied to a vector, and once, when we compute $\widetilde{S}_{\Pi\Pi}$. We will describe the parallel algorithm in Section 3.4.

2.2.2 BDDC Method

Let us define the block matrices

$$K_{\Pi\Pi} = \mathrm{diag}_{i=1}^N(K_{\Pi\Pi}^{(i)}), \quad K_{\Pi I} = \mathrm{diag}_{i=1}^N(K_{\Pi I}^{(i)}), \quad K_{\Pi\Delta} = \mathrm{diag}_{i=1}^N(K_{\Pi\Delta}^{(i)}),$$

and the right hand side vectors

$$f_I^T = [f_I^{(1)T}, \ldots, f_I^{(N)T}], \quad f_\Delta^T = [f_\Delta^{(1)T}, \ldots, f_\Delta^{(N)T}], \quad f_\Pi^T = [f_\Pi^{(1)T}, \ldots, f_\Pi^{(N)T}].$$

We refer to the matrices $K_{II}, K_{\Delta\Delta}$, $K_{\Delta I}$ and R_Δ^B as defined as in (2.3) and (2.4).

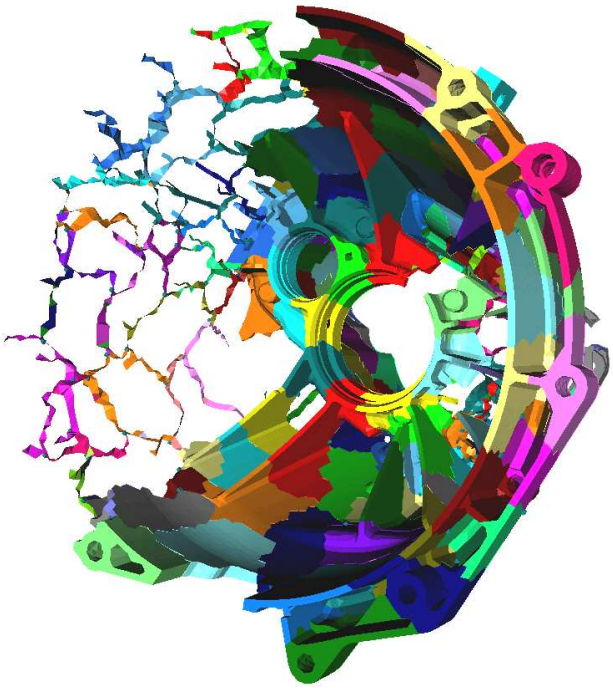

Figure 2.2
The FETI-DP and BDDC methods are Schur complement methods. Unknowns in the interior of subdomains are eliminated upfront (left part of the picture) then the remaining interface problem is solved iteratively by a preconditioned Krylov subspace method.

We first gather all subdomain stiffness matrices in one global blockmatrix

$$\begin{bmatrix} K_{II} & K_{\Delta I}^T & K_{\Pi I}^T \\ K_{\Delta I} & K_{\Delta\Delta} & K_{\Pi\Delta}^T \\ K_{\Pi I} & K_{\Pi\Delta} & K_{\Pi\Pi} \end{bmatrix}.$$

We then eliminate all subdomain interior variables to form the Schur complement

$$\begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \mathcal{S}_{\Pi\Delta}^T \\ \mathcal{S}_{\Pi\Delta} & \mathcal{S}_{\Pi\Pi} \end{bmatrix} = \begin{bmatrix} K_{\Delta\Delta} & K_{\Pi\Delta}^T \\ K_{\Pi\Delta} & K_{\Pi\Pi} \end{bmatrix} - \begin{bmatrix} K_{\Delta I} \\ K_{\Pi I} \end{bmatrix} K_{II}^{-1} \begin{bmatrix} K_{\Delta I}^T K_{\Pi I}^T \end{bmatrix}$$

and corresponding right hand side

$$\begin{bmatrix} g_\Delta \\ g_\Pi \end{bmatrix} = \begin{bmatrix} f_\Delta - K_{\Delta I} K_{II}^{-1} f_I \\ f_\Pi - K_{\Pi I} K_{II}^{-1} f_I \end{bmatrix},$$

where we still have no coupling between the subdomains.

We define the partially assembled Schur complement

$$\widetilde{\mathcal{S}} = \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix} = \begin{bmatrix} I_\Delta & 0 \\ 0 & R_\Pi^T \end{bmatrix} \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \mathcal{S}_{\Pi\Delta}^T \\ \mathcal{S}_{\Pi\Delta} & \mathcal{S}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} I_\Delta & 0 \\ 0 & R_\Pi \end{bmatrix}$$

and the fully assembled Schur complement

$$\mathcal{S} = \begin{bmatrix} R_\Delta^T & 0 \\ 0 & I_\Pi \end{bmatrix} \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} R_\Delta & 0 \\ 0 & I_\Pi \end{bmatrix}.$$

The BDDC preconditioner then is

$$M_{\mathrm{BDDC}}^{-1} = \begin{bmatrix} R_{\Delta,D}^T R_\Delta^B & 0 \\ 0 & I_\Pi \end{bmatrix} \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} (R_\Delta^B)^T R_{\Delta,D} & 0 \\ 0 & I_\Pi \end{bmatrix},$$

where $R_{\Delta,D}$ are scaled variants of the assembly operator R_Δ. As in the FETI-DP jump operator $B_{\Delta,D}$, each entry is scaled by the inverse of its multiplicity.

If we introduce the notation

$$u_\Gamma = \begin{bmatrix} u_\Delta \\ u_\Pi \end{bmatrix}, \quad g_\Gamma = \begin{bmatrix} R_\Delta^T (f_\Delta - K_{\Delta I} K_{II}^{-1} f_I) \\ \tilde{f}_\Pi - \widetilde{K}_{\Pi I} K_{II}^{-1} f_I \end{bmatrix}$$

we can define the BDDC algorithm as the CG method to solve

$$\mathcal{S} u_\Gamma = g_\Gamma$$

with the preconditioner M^{-1}_{BDDC}.

We only note that a BDDC method can be defined which has the same eigenvalues as the FETI-DP method with the lumped preconditioner, see also Li and Widlund [91], by

$$M^{-1}_{\mathrm{BDDC}_L} = \begin{bmatrix} I_I & 0 & 0 \\ 0 & R^T_{\Delta,D} & 0 \\ 0 & 0 & I_\Pi \end{bmatrix} \begin{bmatrix} K_{II} & K^T_{\Delta I} & \widetilde{K}^T_{I\Pi} \\ K_{\Delta I} & K_{\Delta\Delta} & \widetilde{K}^T_{\Pi\Delta} \\ \widetilde{K}_{\Pi I} & \widetilde{K}_{\Pi\Delta} & \widetilde{K}_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} I_I & 0 & 0 \\ 0 & R_{\Delta,D} & 0 \\ 0 & 0 & I_\Pi \end{bmatrix},$$

assuming an ordering of K_g where we have interior variables first and primal variables last.

2.3 Scaling for Heterogeneous Problems

We will discuss the 2D case when we do not make all vertices primal; this discussion is also relevant for edges and vertices in 3D. Generally it would be sufficient to enforce $n-1$ pairs of equality constraints for a node which belongs to n subdomains. Instead, $n \times (n-1)/2$ constraints will be used as indicated in Figure 2.3. This, of course, means that neither the $B^{(i)}$ nor B have full rank.

A small modification of $P_{B,D}$ allows the treatment of coefficient jumps across subdomain boundaries. It is helpful to introduce some notation. For indices i, j of u_B we write $i \sim j$ if a Lagrange multiplier is associated with this pair.

Let the vector ρ store the local coefficients of every component of u_B. Let $i \sim j$, by a Lagrange multiplier λ_k; if u_i belongs to the subdomain Ω_m then the entry $d^{(m)}_{kk}$ of $D^{(m)}$ is

$$d^{(m)}_{kk} = \frac{\rho_j}{\sum_{l\sim i} \rho_l}.$$

The scaled jump operator $B_{B,D}$ is now

$$B_{B,D} := \left(D^{(1)} B^{(1)}_{B,D}, \ldots, D^{(N)} B^{(N)}_{B,D} \right),$$

where the diagonal matrices $D^{(i)}$ contain the scaling factors. If all $\rho_i = 1$ then ρ-scaling reduces to the multiplicity scaling. In the same way we define the scaled BDDC assembly operator

$$R_{\Delta,D} := \left(D^{(1)} R^{(1)}_{\Delta}, \ldots, D^{(N)} R^{(N)}_{\Delta} \right).$$

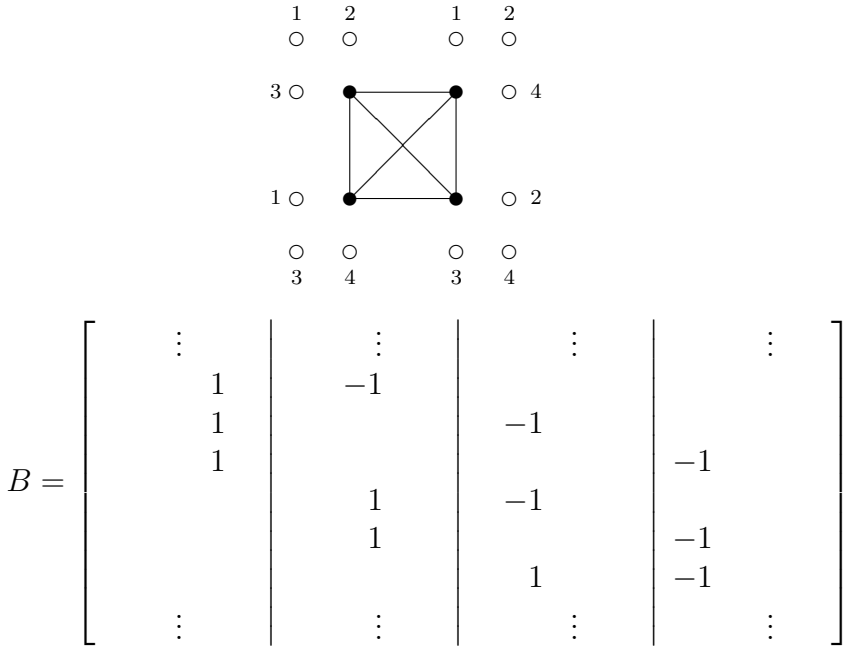

Figure 2.3: *Redundant Lagrange multipliers.*

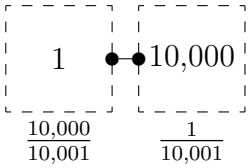

Figure 2.4: *Scaling the contributions from different subdomains.*

2.4 Convergence Estimate for FETI-DP and BDDC

We will first show that FETI-DP and BDDC essentially have the same spectra. The line of argument follows Li and Widlund [92]. Eliminating all interior variables from the FETI-DP system we obtain the system matrix

$$\begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T & B_\Delta^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} & 0 \\ B_\Delta & 0 & 0 \end{bmatrix}.$$

Therefore, we can write the FETI-DP system matrix as

$$F = \begin{bmatrix} B_\Delta & 0 \end{bmatrix} \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} B_\Delta^T \\ 0 \end{bmatrix},$$

and the preconditioner as

$$\begin{aligned} M_{\mathrm{FETI}_D}^{-1} &= B_{\Delta,D} \mathcal{S}_{\Delta\Delta} B_{\Delta,D}^T \\ &= \begin{bmatrix} B_\Delta & 0 \end{bmatrix} \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} B_\Delta^T \\ 0 \end{bmatrix}. \end{aligned}$$

Using the compact notation

$$\widetilde{S} := \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix},$$

$$B_\Gamma := \begin{bmatrix} B_\Delta & 0 \end{bmatrix}, \quad B_{\Gamma,D} := \begin{bmatrix} B_{\Delta,D} & 0 \end{bmatrix},$$

$$P_D := B_{\Gamma,D}^T B_\Gamma,$$

we can write the preconditioned FETI-DP operator as

$$M^{-1}F = B_{D,\Gamma} \widetilde{S} B_{D,\Gamma}^T B_\Gamma \widetilde{S}^{-1} B_\Gamma^T. \tag{2.6}$$

The operator $M^{-1}F$ then has the same eigenvalues as

$$P_D^T \widetilde{S} P_D \widetilde{S}^{-1}. \tag{2.7}$$

It is an important property of the projection P_D that it preserves the jump in the sense that

$$B_\Gamma P_D \tilde{u}_\Gamma = B_\Gamma \tilde{u}_\Gamma \qquad \forall \tilde{u}_\Gamma \in \widetilde{W}_\Gamma. \tag{2.8}$$

This can easily be seen by the following considerations. We recall the equivalence relation $\sim$ over component indices of $u_\Gamma = [u_\Delta^T, \tilde{u}_\Pi^T]^T$, having $i \sim j$ if u_i and u_j share a Lagrange multiplier. Clearly,

$$\begin{aligned} P_D\, \tilde{u}_\Gamma = B_{\Gamma,D}^T B_\Gamma \tilde{u}_\Gamma &= \begin{bmatrix} B_\Gamma^{(1)T} D^{(1)} \\ \vdots \\ B_\Gamma^{(n)T} D^{(n)} \end{bmatrix} \begin{bmatrix} u_{i_1} - u_{j_1} \\ \vdots \\ u_{i_\nu} - u_{j_\nu} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{k_1 \sim 1} d_{k_1} (u_1 - u_{k_1}) \\ \vdots \\ \sum_{k_n \sim n} d_{k_n} (u_n - u_{k_n}) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} B_{\Delta,D}^T B_\Delta u_\Delta \\ 0 \end{bmatrix}. \end{aligned}$$

Here, we have associated the scaling factors with the degree of freedom u_{k_l}, i.e. we define

$$d_{k_l} := \frac{\rho_{k_l}}{\sum_{m\sim k_l}\rho_m}.$$

This is the same scaling as above but this numbering is more convenient for the componentwise notation in the following. It is not helpful in the implementation. We now consider the ith component of the dual part, it holds that

$$\begin{aligned}\left(B_{D,\Delta}^T B_\Delta\, u_\Delta\right)_i &= \sum_{k_i\sim i} d_{k_i}\,(u_i - u_{k_i}) && (2.9)\\ &= u_i - \sum_{k_i\sim i} d_{k_i}\,u_{k_i} && (2.10)\\ &= u_i - (R_\Delta^T R_{\Delta,D} u_\Delta)_i && (2.11)\end{aligned}$$

since $\sum_{k_i\sim i} d_{k_i} = 1\ \forall i$.

We see that we substract from all dual variables the D-weighted average $(R_\Delta^T R_{\Delta,D} u_\Delta)_i$, which is a continuous function. The operator P_D therefore preserves jumps, and we also have $P_D P_D \tilde{u}_\Gamma = P_D \tilde{u}_\Gamma$.

By using the notation

$$R_\Gamma := \begin{bmatrix} R_\Delta & 0 \\ 0 & I_\Pi \end{bmatrix}, \quad R_{\Gamma,D} := \begin{bmatrix} R_{\Delta,D} & 0 \\ 0 & I_\Pi \end{bmatrix},$$

we can define a second projection

$$\begin{aligned}E_D &:= (I - P_D) && (2.12)\\ &= R_\Gamma R_{\Gamma,D}^T, && (2.13)\end{aligned}$$

which then immediately gives

$$E_D + P_D = I, \quad P_D E_D = E_D P_D = 0.$$

The central part of the theory is given by the following theorem, cf. [84].

Theorem 2.4.1 *For all $\tilde{u}_\Gamma \in \widetilde{W}$, we have a bound*

$$|P_D \tilde{u}_\Gamma|_{\tilde{S}}^2 \leq C(1 + \log(H/h))^2 |\tilde{u}_\Gamma|_{\tilde{S}}^2, \tag{2.14}$$

where C is independent of H, h, and the value of ρ_i.

Proof:

We will denote the local Schur complement obtained by eliminating all interior variables of a subdomain Ω_i by

$$S^{(i)} := \begin{bmatrix} \mathcal{S}_{\Delta\Delta}^{(i)} & \mathcal{S}_{\Pi\Delta}^{(i)T} \\ \mathcal{S}_{\Pi\Delta}^{(i)} & \mathcal{S}_{\Pi\Pi}^{(i)} \end{bmatrix} = \begin{bmatrix} K_{\Delta\Delta}^{(i)} & K_{\Pi\Delta}^{(i)T} \\ K_{\Pi\Delta}^{(i)} & K_{\Pi\Pi}^{(i)} \end{bmatrix} - \begin{bmatrix} K_{\Delta I}^{(i)} \\ K_{\Pi I}^{(i)} \end{bmatrix} K_{II}^{(i)-1} \begin{bmatrix} K_{\Delta I}^{(i)T} K_{\Pi I}^{(i)T} \end{bmatrix}.$$

For convenience we also define the global, unassembled Schur complement matrix

$$S := \mathrm{diag}_{i=1}^{N}(S^{(i)}).$$

We also keep in mind that the finite element function $\tilde{u}_\Gamma \in \widetilde{W}$ corresponds to a vector $[u_\Delta^T, \tilde{u}_\Pi^T]^T$. We introduce an operator $R_{\partial\Omega_i}$ which restricts the finite element function $\tilde{u}_\Gamma \in \widetilde{W}_\Gamma$ to the subdomain boundary $\partial\Omega_i$. The matrix representation is defined by

$$\begin{bmatrix} u_\Delta^{(i)} \\ u_\Pi^{(i)} \end{bmatrix} = R_{\partial\Omega_i} \begin{bmatrix} u_\Delta \\ \tilde{u}_\Pi \end{bmatrix}.$$

For all $\tilde{u}_\Gamma \in \widetilde{W}_\Gamma$, i.e. functions which satisfy the primal constraints, it holds

$$|\tilde{u}_\Gamma|_{\tilde{S}}^2 = \sum_{i=1}^{N} |R_{\partial\Omega_i}\tilde{u}_\Gamma|_{S^{(i)}}^2,$$

where $R_{\partial\Omega_i}\tilde{u}_\Gamma$ is the restriction of $\tilde{u}_\Gamma$ to $\partial\Omega_i$.

Thus, (2.14) becomes

$$\sum_{i=1}^{N} |R_{\partial\Omega_i} P_D \tilde{u}_\Gamma|_{S^{(i)}}^2 \leq C_1 \sum_{i=1}^{N} |R_{\partial\Omega_i}\tilde{u}_\Gamma|_{S^{(i)}}^2.$$

We can establish the global estimate on P_D by bounds that can be calculated locally.

Our finite element functions $u \in H^1(\Omega_i)$ are continuous whereas we do not have a Sobolev inequality in 2D for functions in $H^1(\Omega_i)$. But for finite element functions we have a discrete Sobolev inequality, see Toselli and Widlund [136, Lemma 4.15], Brenner and Scott [19, Theorem 4.9.1], or similarly Bramble, Pasciak, and Schatz [18, Lemma 3.5],

$$\|u\|_{L_\infty(\Omega_i)}^2 \leq C(1 + \log(H/h))\|u\|_{H^1(\Omega_i)}^2; \tag{2.15}$$

the L_2-term on the right hand side can be removed by subtracting from u any convex combination of u. The original proof of the condition number bound by Mandel and Tezaur [100] used the formulation of Bramble, Pasciak and Schatz [18].

We will now show a condition number estimate, which follows the presentation for three dimensional problems in Klawonn, Widlund, and Dryja [86] and Klawonn and Widlund [84].

Let $\Theta_{\mathcal{E}^{ij}}$ and $\Theta_{\mathcal{V}^{ik}}$ be the characteristic finite element functions associated with an edge $\mathcal{E}^{ij}$ and a vertex $\mathcal{V}^{ik}$. These functions take the value one at the

nodes of the edge or at the vertex and zero elsewhere. The functions form a partition-of-unity associated with the decomposition of the interface Γ into edges and vertices. If I_h denotes the finite element interpolation operator, we obtain

$$v^{(i)} = \sum_{\mathcal{E}^{ij}} I_h(\Theta_{\mathcal{E}^{ij}} v^{(i)}) + \sum_{\mathcal{V}^{ik}} I_h(\Theta_{\mathcal{V}^{ik}} v^{(i)}) \tag{2.16}$$

for a finite element function $v^{(i)}$ local to a subdomain boundary $\partial\Omega_i$. We will use this decomposition for $v^{(i)} = R_{\partial\Omega_i} P_D u_\Gamma$ and find bounds for the contributions from the edges

$$\rho_i \| I_h(\Theta_{\mathcal{E}^{ij}} v^{(i)}) \|^2_{H^{1/2}(\partial\Omega_i)} \tag{2.17}$$

and the vertices

$$\rho_i \| I_h(\Theta_{\mathcal{V}^{ij}} v^{(i)}) \|^2_{H^{1/2}(\partial\Omega_i)}. \tag{2.18}$$

These bounds will be in terms of $|R_{\partial\Omega_i} \tilde{u}_\Gamma|^2_{S^{(i)}}$. By using the triangle inequality we are able to establish the global bound. Every contribution only appears a bounded number of times in the global sum if every subdomain has a bounded number of faces. In the contributions (2.17), (2.18) we use full norms; we can remove the L_2-terms later by using the continuity at the primal vertices.

For the edge contributions we use

$$|I_h(\Theta_{\mathcal{E}^{ij}} u)|^2_{H^{1/2}(\partial\Omega)} \leq C \left(|u|^2_{H^{1/2}(\partial\Omega)} + (1 + \log(H/h)) \|u\|^2_{L_\infty(\partial\Omega)} \right), \tag{2.19}$$

see, e.g., Brenner and Scott [19, Theorem 7.5.30].

Since our finite element functions are continuous, we have

$$\|u\|^2_{L_\infty(\partial\Omega_i)} \leq \|\mathcal{H}u\|^2_{L_\infty(\overline{\Omega_i})} = \|\mathcal{H}u\|^2_{L_\infty(\Omega_i)}.$$

Hence, for a finite element function u on the edge $\mathcal{E}^{ij}$ we have from (2.19) and (2.15)

$$\begin{aligned} \|I_h(\Theta_{\mathcal{E}^{ij}} u)\|^2_{H^{1/2}(\partial\Omega)} &\leq C(1 + \log(H/h))^2 \|\mathcal{H}u\|^2_{H^1(\Omega)} \\ &= C(1 + \log(H/h))^2 \|u\|^2_{H^{1/2}(\Omega)}. \end{aligned} \tag{2.20}$$

We can remove the L_2-term on the right hand side if u has edge average zero, i.e. $\overline{u}_{\mathcal{E}^{ij}} = 0$, by using a Poincaré inequality.

If all vertices are primal, the vertex contributions vanish since P_D restricted to the primal variables is the zero mapping. We therefore only need to estimate the edge contributions. We will also need the property of the scaling

$$\rho_i d_j^2 = \rho_i \frac{\rho_j^2}{(\rho_i + \rho_j)^2} \leq \min\{\rho_i, \rho_j\}.$$

Using (2.9), the edge contribution is

$$\begin{aligned}
& \rho_i \|I_h(\Theta_{\mathcal{E}^{ij}} d^{(i)}(w^{(i)} - w^{(j)}))\|^2_{H^{1/2}(\partial\Omega_i)} \\
= \;& \rho_i \|I_h(\Theta_{\mathcal{E}^{ij}} d^{(i)}(w^{(i)} - \overline{w}_{i,\mathcal{E}^{ij}}) - (w^{(j)} - \overline{w}_{j,\mathcal{E}^{ij}}) + (\overline{w}_{i,\mathcal{E}^{ij}} - \overline{w}_{j,\mathcal{E}^{ij}})))\|^2_{H^{1/2}(\partial\Omega_i)} \\
= \;& 2\min\{\rho_i,\rho_j\} \|I_h(\Theta_{\mathcal{E}^{ij}}((w^{(i)} - \overline{w}_{i,\mathcal{E}^{ij}}) - (w^{(j)} - \overline{w}_{j,\mathcal{E}^{ij}})))\|^2_{H^{1/2}(\partial\Omega_i)} \\
& + 2\min\{\rho_i,\rho_j\} \|\Theta_{\mathcal{E}^{ij}}(\overline{w}_{i,\mathcal{E}^{ij}} - \overline{w}_{j,\mathcal{E}^{ij}})\|^2_{H^{1/2}(\partial\Omega_i)}
\end{aligned}$$

We have added and subtracted the edge averages so that we can estimate the first term using (2.20) and a Poincaré inequality,

$$\begin{aligned}
& \|I_h(\Theta_{\mathcal{E}^{ij}} \left((w^{(i)} - \overline{w}_{i,\mathcal{E}^{ij}}) - (w^{(j)} - \overline{w}_{j,\mathcal{E}^{ij}})\right) \|^2_{H^{1/2}(\partial\Omega_i)} \\
\leq \;& \|I_h(\Theta_{\mathcal{E}^{ij}}(w^{(i)} - \overline{w}_{i,\mathcal{E}^{ij}}))\|^2_{H^{1/2}(\partial\Omega_i)} + \|I_h(\Theta_{\mathcal{E}^{ij}}(w^{(j)} - \overline{w}_{j,\mathcal{E}^{ij}}))\|^2_{H^{1/2}(\partial\Omega_i)} \\
\leq \;& C(1+\log(H/h))^2 \left(|w^{(i)}|^2_{H^{1/2}(\partial\Omega_i)} + |w^{(j)}|^2_{H^{1/2}(\partial\Omega_j)}\right).
\end{aligned}$$

It remains to estimate the term

$$\|\Theta_{\mathcal{E}^{ij}}(\overline{w}_{i,\mathcal{E}^{ij}} - \overline{w}_{j,\mathcal{E}^{ij}})\|^2_{H^{1/2}(\partial\Omega_i)} = (\overline{w}_{i,\mathcal{E}^{ij}} - \overline{w}_{j,\mathcal{E}^{ij}})^2 \|\Theta_{\mathcal{E}^{ij}}\|^2_{H^{1/2}(\partial\Omega_i)}.$$

From (2.19) with $u = 1$ we obtain an estimate for the energy of the characteristic function of the edge,

$$\|\Theta_{\mathcal{E}^{ij}}\|^2_{H^{1/2}(\partial\Omega_i)} \leq C(1+\log(H/h)).$$

Adding and subtracting the value

$$w_{i,\mathcal{V}^{ij}} = w_{j,\mathcal{V}^{ij}} \tag{2.21}$$

at a primal vertex $\mathcal{V}^{ij} \in \partial\mathcal{E}^{ij}$, we get

$$(\overline{w}_{i,\mathcal{E}^{ij}} - \overline{w}_{j,\mathcal{E}^{ij}})^2 \leq 2(\overline{w}_{i,\mathcal{E}^{ij}} - w_{i,\mathcal{V}^{ij}})^2 + 2(\overline{w}_{j,\mathcal{E}^{ij}} - w_{j,\mathcal{V}^{ij}})^2.$$

Note that the expressions on the right hand side, e.g.

$$(\overline{w}_{i,\mathcal{E}^{ij}} - w_{i,\mathcal{V}^{ij}})^2,$$

are shift invariant, i.e. they do not change their value if we add or substract a constant to $w^{(i)}$ or $w^{(j)}$.

Therefore, we can assume that $\overline{w}_{i,\mathcal{E}^{ij}} = \overline{w}_{j,\mathcal{E}^{ij}} = 0$, and it remains to estimate $(w_{i,\mathcal{V}^{ij}})^2$ and $(w_{j,\mathcal{V}^{ij}})^2$. From (2.15), we obtain

$$\begin{aligned}
|w_{i,\mathcal{V}^{ij}}|^2 &\leq \|w^{(i)}\|^2_{L_\infty(\partial\Omega_i)} \\
&\leq \|\mathcal{H}w^{(i)}\|^2_{L_\infty(\Omega_i)} \\
&\leq C(1+\log(H/h))\|\mathcal{H}w^{(i)}\|^2_{H^1(\Omega)} \\
&= C(1+\log(H/h))\|w^{(i)}\|^2_{H^{1/2}(\partial\Omega_i)} \\
&\leq C(1+\log(H/h))|w^{(i)}|^2_{H^{1/2}(\partial\Omega_i)}.
\end{aligned} \tag{2.22}$$

In the last inequality we have removed the L_2-term by a Poincaré inequality on $\partial\Omega_i$, again using $\overline{w}_{i,\mathcal{E}^{ij}} = 0$. If we have chosen all vertices as primal then the proof is completed.

If a vertex $\mathcal{V}^{ij}$ is not primal, we have to estimate the contributions from this vertex, see (2.16) and (2.18). For simplicity, let us assume that every pair of neighboring subdomains share an edge, i.e. the multiplicity of all vertices is less than four. Interestingly, irregular partitions of meshes rarely have vertices with a multiplicity of four or higher. We have

$$\begin{aligned} & \rho_i \|\Theta_{\mathcal{V}^{ij}} v^{(i)}(\mathcal{V}^{ij})\|^2_{H^{1/2}(\partial\Omega)} \\ \leq\ & C \sum_{l \in \text{neighb}(i,\mathcal{V}^{ij})} \rho_i d_l \|\Theta_{\mathcal{V}^{ij}}\|^2_{H^{1/2}(\partial\Omega_i)} (w_{i,\mathcal{V}^{ij}} - w_{l,\mathcal{V}^{ij}})^2 \end{aligned} \tag{2.23}$$

From a simple computation, see also [19, Theorem 7.5.31] and [136, Lemma B.5], we have

$$\|\Theta_{\mathcal{V}^{ij}}\|^2_{H^{1/2}(\partial\Omega_i)} \leq C$$

with a constant independent of h and H. To estimate $(w_{i,\mathcal{V}^{ij}} - w_{l,\mathcal{V}^{ij}})^2$ we assume that the other vertex $\mathcal{V}^{ik}$ adjacent to the relevant edge is primal, i.e.

$$w_{i,\mathcal{V}^{ik}} = w_{l,\mathcal{V}^{ik}}.$$

Adding and subtracting the value at the primal vertex, we obtain

$$\begin{aligned} & (w_{i,\mathcal{V}^{ij}} - w_{l,\mathcal{V}^{ij}})^2 \\ \leq\ & 2(w_{i,\mathcal{V}^{ij}} - w_{i,\mathcal{V}^{ik}})^2 + 2(w_{l,\mathcal{V}^{ij}} - w_{l,\mathcal{V}^{ik}})^2. \end{aligned} \tag{2.24}$$

By inserting the (discontinuous) edge averages we get

$$\begin{aligned} & (w_{i,\mathcal{V}^{ij}} - w_{l,\mathcal{V}^{ij}})^2 \\ \leq\ & 4(w_{i,\mathcal{V}^{ij}} - \overline{w}_{i,\mathcal{E}^{ij}})^2 + 4(w_{i,\mathcal{V}^{ik}} - \overline{w}_{i,\mathcal{E}^{ij}})^2 \\ & + 4(w_{l,\mathcal{V}^{ij}} - \overline{w}_{l,\mathcal{E}^{ij}})^2 + 4(w_{l,\mathcal{V}^{ik}} - \overline{w}_{l,\mathcal{E}^{ij}})^2. \end{aligned} \tag{2.25}$$

Here, each expression is again shift invariant and we can assume that $\overline{w}_{i,\mathcal{E}^{ij}} = \overline{w}_{l,\mathcal{E}^{ij}} = 0$. We can continue as in (2.22) to get

$$\begin{aligned} & 2(w_{i,\mathcal{V}^{ij}})^2 + 2(w_{l,\mathcal{V}^{ij}}))^2 \\ \leq\ & C(1 + \log(H/h))\big(|w^{(i)}|^2_{H^{1/2}(\partial\Omega_i)} + |w^{(j)}|^2_{H^{1/2}(\partial\Omega_i)}\big). \end{aligned} \tag{2.26}$$

We have removed the L_2-term by a Poincaré inequality. This concludes the bound under the assumption made above.

If we further reduce the number of primal vertices, we may have the situation that for an edge $\mathcal{E}^{ij}$ there is no adjacent primal vertex. In that case (2.21) does not hold. Also, we may want to consider vertices with a

multiplicity of four or higher. Under certain conditions, we can still derive a bound. For simplicity, let us assume a homogeneous material with no jumps in the coefficients. We define two subdomains as connected in a graph $G_{\mathcal{V}}$ if they share a primal vertex. In order to establish the desired bound for an edge or a vertex contribution we need to demand that every subdomain is connected in $G_{\mathcal{V}}$ to all its neighbors, possibly passing through several other subdomains. In that case we add and subtract all primal vertices along the path and have to estimate each term in the same way as in (2.24). For the vertex contributions this gives us a bound of the form

$$(1+\log(H/h))\left(|w^{(i)}|^2_{H^{1/2}(\partial\Omega_i)}+|w^{(l)}|^2_{H^{1/2}(\partial\Omega_l)}+\sum_{k\in\text{path}}|w^{(k)}|_{H^{1/2}(\partial\Omega_k)}\right).$$

For the edge contributions we get the same bound with two powers of the logarithmic factor. Here, whenever a path passes through a subdomain its contribution is added to the estimate and the condition number of the resulting algorithm will grow. Still, asymptotically, we get a good bound of the desired polylogarithmic form if the length of the path is uniformly bounded by a constant.

This argument can be extended to the case of discontinuous coefficients, using the concept of an acceptable path; see [84] and [86].

□

Using this theorem, we can prove Theorem 2.4.2 as given in Klawonn and Widlund [84].

Theorem 2.4.2 *The condition number of the preconditioned FETI-DP operator satisfies the bound*

$$\kappa(M^{-1}F)\leq C(1+\log(H/h))^2.$$

where $C_1=C(1+\log(H/h))^2$.

Proof: We will show the following bounds given in the energy norm defined by F,

$$\langle\lambda,\lambda\rangle_F\leq\left\langle M^{-1}F\lambda,\lambda\right\rangle_F\leq C(1+\log(H/h))^2\langle\lambda,\lambda\rangle_F.$$

This will give us a lower bound of one on the lowest eigenvalue and the desired bound on the condition number. We can establish the lower bound

using (2.8). We have for all λ,

$$\begin{aligned}
\langle \lambda, \lambda \rangle_F &= \left\langle B_\Gamma \widetilde{S}^{-1} B_\Gamma^T \lambda \,, \lambda \right\rangle \\
&= \left\langle B_\Gamma \widetilde{S}^{-1} P_D^T B_\Gamma^T \lambda \,, \lambda \right\rangle \\
&= \left\langle B_\Gamma \widetilde{S}^{-1} B_\Gamma^T B_{D,\Gamma} \widetilde{S}^{1/2} \widetilde{S}^{-1/2} B_\Gamma^T \lambda \,, \lambda \right\rangle \\
&= \left\langle \widetilde{S}^{-1/2} B_\Gamma^T \lambda \,, \widetilde{S}^{1/2} B_{D,\Gamma}^T B_\Gamma \widetilde{S}^{-1} B_\Gamma^T \lambda \right\rangle \\
&\leq \left\langle \widetilde{S}^{-1/2} B_\Gamma^T \lambda \,, \widetilde{S}^{-1/2} B_\Gamma^T \lambda \right\rangle \left\langle \widetilde{S}^{1/2} B_{D,\Gamma}^T B_\Gamma \widetilde{S}^{-1} B_\Gamma^T \lambda \,, \widetilde{S}^{1/2} B_{D,\Gamma}^T B_\Gamma \widetilde{S}^{-1} B_\Gamma^T \lambda \right\rangle \\
&= \langle \lambda, \lambda \rangle_F \left\langle M^{-1} F \lambda, \lambda \right\rangle_F .
\end{aligned}$$

The upper bound can be established by using Theorem 2.4.1,

$$\begin{aligned}
\left\langle M^{-1} F \lambda, \lambda \right\rangle_F &= \left\langle M^{-1} F \lambda \,, F \lambda \right\rangle \\
&= \left\langle B_{D,\Gamma}^T F \lambda \,, B_{D,\Gamma}^T F \lambda \right\rangle_{\widetilde{S}} \\
&= \left\langle P_D \widetilde{S}^{-1} B_\Gamma^T \lambda \,, P_D \widetilde{S}^{-1} B_\Gamma^T \lambda \right\rangle_{\widetilde{S}} \\
&\leq C(1 + \log(H/h))^2 \left\langle \widetilde{S}^{-1} B_\Gamma^T \lambda \,, \widetilde{S}^{-1} B_\Gamma^T \lambda \right\rangle_{\widetilde{S}} \\
&\leq C(1 + \log(H/h))^2 \left\langle B_\Gamma \widetilde{S}^{-1} B_\Gamma^T \,, \lambda \right\rangle \\
&\leq C(1 + \log(H/h))^2 \langle \lambda, \lambda \rangle_F
\end{aligned}$$

□

We can write the BDDC system matrix as

$$\mathcal{S} = \begin{bmatrix} R_\Delta^T \mathcal{S}_{\Delta\Delta} R_\Delta & R_\Delta^T \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} R_\Delta & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix}$$

and the BDDC preconditioner in the form

$$\begin{aligned}
& M_{\mathrm{BDDC}}^{-1} \\
= & \begin{bmatrix} R_{D,\Delta}^T & 0 \\ 0 & I_\Pi \end{bmatrix} \begin{bmatrix} \mathcal{S}_{\Delta\Delta} & \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ \widetilde{\mathcal{S}}_{\Pi\Delta} & \widetilde{\mathcal{S}}_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} R_{D,\Delta} & 0 \\ 0 & I_\Pi \end{bmatrix} \\
= & \begin{bmatrix} I & -R_{D,\Delta}^T \mathcal{S}_{\Delta\Delta}^{-1} \widetilde{\mathcal{S}}_{\Pi\Delta}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} R_{D,\Delta}^T \mathcal{S}_{\Delta\Delta}^{-1} R_{D,\Delta} & 0 \\ 0 & \widetilde{S}_{\Pi\Pi}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\widetilde{\mathcal{S}}_{\Pi\Delta} \mathcal{S}_{\Delta\Delta}^{-1} R_{D,\Delta} & I \end{bmatrix}.
\end{aligned}$$

Here, it becomes obvious that the global coupling in the BDDC method is also given by the Schur complement matrix $\widetilde{S}_{\Pi\Pi}$, which also represents the FETI-DP coarse problem.

Using the operators R_Γ and $R_{\Gamma,D}$, the product $M_{\mathrm{BDDC}}^{-1}\mathcal{S}$ becomes

$$R_{\Gamma,D}^T \widetilde{S}^{-1} R_{\Gamma,D} R_\Gamma^T \widetilde{S} R_\Gamma, \tag{2.27}$$

which has the same eigenvalues as

$$E_D \widetilde{S}^{-1} E_D^T \widetilde{S}. \tag{2.28}$$

Together with Formulas (2.6) and (2.12) this can be used to construct an elegant and short proof that the eigenvalues of FETI-DP and BDDC algorithms are basically the same.

Theorem 2.4.3 *The FETI-DP and the BDDC method share all eigenvalues except, possibly, for eigenvalues equal to zero and one.*

Proof: Considering the eigenvalues of (2.7) and (2.28) and using (2.12) it can be shown by linear algebra only that the spectra of FETI-DP and BDDC are essentially the same; see Li and Widlund [92, Section 5] for further details.

□

This theorem ensures that the bounds established for the FETI-DP algorithm also apply to the BDDC method. The first proof of this fact was given by Mandel, Dohrmann and Tezaur [98].

Chapter 3

FETI-DP for Linear Elasticity

3.1 Equations of Linear Elasticity

The equations of linear elasticity model the displacement of an elastic material under the action of external and internal forces. The elastic body occupies a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. We denote its boundary by $\partial\Omega$ and assume that one part of it, $\partial\Omega_D$, is clamped, e.g. with homogeneous Dirichlet boundary conditions, and that the rest, $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$, is subject to a surface force g, i.e. a natural boundary condition. We can also introduce a body force f, e.g. gravity.

We measure the deformation of the body Ω by the (linearized) strain tensor

$$\epsilon_{ij} = \frac{1}{2}\left(\partial_j u_i + \partial_i u_j\right),$$

where $u \in \mathbf{R}^d$ is the displacement vector. The deformation results in stresses $\sigma(\epsilon)$ within the body. We call a body linear elastic if the stress is a linear function of the strain, i.e.

$$\sigma = \mathbf{C}\epsilon.$$

The elasticity tensor $\mathbf{C}$ models the material and depends on the local properties of the body.

The equilibrium of forces is then given by the coupled system

$$\begin{aligned}
\sum_{j=1}^{d} \partial_j \sigma_{ij}(u) &= f_i \quad \text{in } \Omega, \\
\sigma_{ij}(u) &= \sum_{k,l=1}^{d} \mathbf{c}_{ijkl}\, \epsilon_{kl}(u), \\
\epsilon_{ij} &= \frac{1}{2}\left(\partial_j u_i + \partial_i u_j\right), \\
u &= 0 \quad \text{on } \partial\Omega_D, \\
\sum_{j=1}^{d} \sigma_{ij}(u)\nu_j &= g_i \quad \text{on } \partial\Omega_N,
\end{aligned}$$

which can also be written

$$\operatorname{div}\sigma = f \text{ in } \Omega, \quad \sigma = \mathbf{C}\epsilon(u), \quad \epsilon = \frac{1}{2}(\nabla u + (\nabla u)^T),$$

$$u = 0 \text{ on } \partial\Omega_D, \quad \langle \sigma, \nu \rangle = g \text{ on } \partial\Omega_N.$$

By multiplying with $v_i \in H_0^1(\Omega, \partial\Omega_N)$, assuming symmetry of σ_{ij}, and using a Green's formula,

$$\int_\Omega \partial_j \sigma_{ij}\, v_i \, dx = -\int_\Omega \sigma_{ij}\, \partial_j v_i \, dx + \int_{\partial\Omega_N} \sigma_{ij}\, v_i\, \nu_j \, ds,$$

we can derive the weak formulation

$$a(u, v) = f(v) \quad \forall v \in V,$$

where

$$a(u, v) = \int_\Omega \sum_{i,j=1}^{d} \sigma_{ij}(u)\epsilon_{ij}(v)\, dx =: \int_\Omega \sigma(u) : \epsilon(v)\, dx, \tag{3.1}$$

$$f(v) = \int_\Omega \sum_{i=1}^{d} f_i v_i \, dx + \int_{\partial\Omega_N} \sum_{i=1}^{d} g_i v_i \, ds = \int_\Omega \langle f, v \rangle \, dx + \int_{\partial\Omega_N} \langle g, v \rangle \, ds,$$

in the appropriate Sobolov space

$$V = \mathbf{H_0^1}(\Omega, \partial\Omega_D) := (H_0^1(\Omega, \partial\Omega_D))^d.$$

Under the assumption that the material is isotropic, i.e. the response to a force is independent of the orientation, we have that

$$\sigma = 2\mu\epsilon + \lambda \text{trace}\,(\epsilon) I,$$

or written componentwise

$$\sigma_{ij} = 2\mu\epsilon_{ij}(u) + \lambda\delta_{ij}\sum_{k=1}^{d}\epsilon_{kk}(u), \tag{3.2}$$

where δ_{ij} is the Kronecker symbol. The constants $\lambda \geq 0$ and $\mu > 0$ are referred to as the Lamé parameters.

Inserting (3.2) into (3.1), we derive a simplified weak formulation,

$$2\mu\int_\Omega \sum_{i,j=1}^{d}\epsilon_{ij}\epsilon_{ij}\,dx + \lambda\int_\Omega (\sum_{i=1}^{d}\partial_i v_i)(\sum_{i=1}^{d}\partial_i v_i) = \int_\Omega \sum_{i=1}^{d} f_i v_i + \int_{\partial\Omega_N}\sum_{i=1}^{d} g_i v_i,$$

which we can also write

$$2\mu\int_\Omega \epsilon(u) : \epsilon(v)\,dx + \lambda\int_\Omega \text{div}\,(u)\text{div}\,(v)\,dx = \int_\Omega \langle f, v\rangle\,dx + \int_{\partial\Omega_N}\langle g, v\rangle\,ds.$$

Instead of the Lamé parameters, one can also derive formulations using the Young's modulus E and Poisson's ratio ν. We have the relations

$$E = \frac{\mu(2\mu+3\lambda)}{\mu+\lambda}, \quad \nu = \frac{\lambda}{2(\mu+\lambda)}$$

or

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

For convenience, we define

$$\langle \mathbf{F}, v\rangle = \left(\int_\Omega \langle f, v\rangle\,dx + \int_{\partial\Omega_N}\langle g, v\rangle\,ds\right).$$

In the following, we choose the notation

$$\int_\Omega G(x)\,\epsilon(u) : \epsilon(v)\,dx + \int_\Omega G(x)\,\beta(x)\,\text{div}\,(u)\text{div}\,(v)\,dx = \langle \mathbf{F}, v\rangle\,, \tag{3.3}$$

where $G(x) = 2\mu$ and $\beta(x) = \lambda/(2\mu)$.

In order to show unique solvability in the space $\mathbf{H_0^1}(\Omega, \partial\Omega_D)$, the ellipticity and continuity of

$$\begin{aligned} a(u,v) &= \int_\Omega G(x)\epsilon(u) : \epsilon(v)\, dx + \int_\Omega G(x)\beta(x)\mathrm{div}\,(u)\mathrm{div}\,(v)\, dx \\ &= (G\,\epsilon(u), \epsilon(v))_{L_2(\Omega)} \;+\; (G\,\beta\,\mathrm{div}\, u, \mathrm{div}\, v)_{L_2(\Omega)} \end{aligned}$$

as well as the continuity of $\langle \mathbf{F}, v\rangle$ has to be established.

Continuity follows from elementary considerations whereas ellipticity is nontrivial and follows from Korn's second inequality,

$$C(\Omega)\|u\|^2_{H^1(\Omega)} \leq (\epsilon(u) : \epsilon(u))_{L_2(\Omega)} + \|u\|^2_{L_2(\Omega)};$$

see, e.g., [25, 19].

Here, $C(\Omega) > 0$ depends on the domain Ω. As a consequence of this inequality we have Korn's first inequality

$$c\|u\|^2_{H^1(\Omega)} \leq (\epsilon, \epsilon)_{L_2(\Omega)} \leq C\|u\|^2_{H^1(\Omega)}$$

for all u with essential boundary conditions, i.e. for $u \in \mathbf{H_0^1}(\Omega, \partial\Omega_D)$ and $(\int_{\partial\Omega_D} 1\, ds)^2 > 0$.

The null space $\mathbf{ker}\,(\epsilon)$ of ϵ is the space of the rigid body motions. In two dimensions it is spanned by two translations,

$$r_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{3.4}$$

and one rotation (or the linear approximation to the rotation),

$$r_3 = \begin{bmatrix} -(x_1 - \hat{x}_1) \\ x_2 - \hat{x}_2 \end{bmatrix}. \tag{3.5}$$

Here, we have shifted the origin of the rotation to the point $\hat{x} \in \Omega$.

In three space dimensions the null space is spanned by three translations,

$$r_1 := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad r_2 := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad r_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.6}$$

and three rotations,

$$r_4 := \begin{bmatrix} x_2 - \hat{x}_2 \\ -(x_1 - \hat{x}_1) \\ 0 \end{bmatrix}, \; r_5 := \begin{bmatrix} -(x_3 - \hat{x}_3) \\ 0 \\ x_1 - \hat{x}_1 \end{bmatrix}, \; r_6 := \begin{bmatrix} 0 \\ x_3 - \hat{x}_3 \\ -(x_2 - \hat{x}_2) \end{bmatrix}. \tag{3.7}$$

We have shifted the origin of the rotation to the point $\hat{x} \in \Omega$. From the Korn inequalities, we see that we can control the null space of the operator if we set essential boundary conditions or if we require the solution to be orthogonal to all rigid body modes.

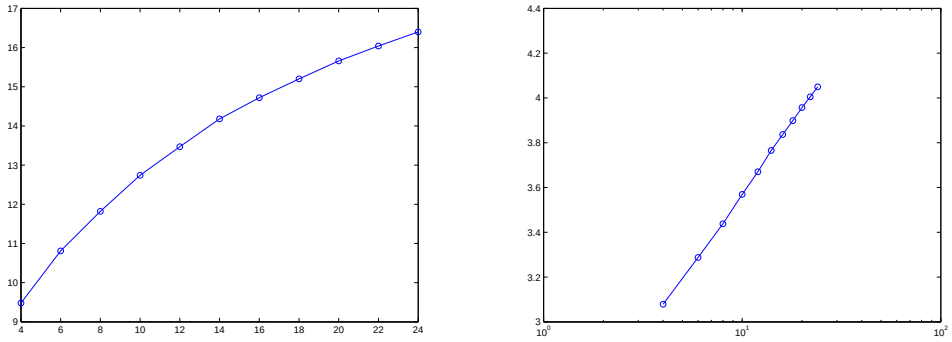

Figure 3.1
Condition number of the FETI-DP operator for 2D linear elasticity using vertex constraints plotted versus H/h. Left: Plot of κ versus H/h. Right: Semilogarithmic plot of $\sqrt{\kappa}$ versus H/h.

3.2 Need for an Appropriate Coarse Space

We can apply our formulation of the FETI-DP (and BDDC) algorithm, given for scalar equations in Chapter 2, without changes also to systems of PDEs. We only have to specify how to choose the primal variables, i.e. we have to state how to define the coarse space. In elasticity, the dimension of the null space of the operator is three in 2D and six in 3D. In scalar equations one primal constraint per subdomain was sufficient to ensure invertibility of $\widetilde{K}$. Moreover, to establish a condition number bound of the form

$$\kappa(M^{-1}F) \leq C(1 + \log(H/h))^2,$$

e.g. one vertex per edge is sufficient, see Chapter 2.

Both these results do not hold for the case of elasticity anymore. An obvious choice is still to choose vertex unknowns as the primal variables. If we have a sufficient number of vertices per subdomain, we can control the rigid body motions of each subdomain. We have invertibility of $\widetilde{K}$ if the closure of every subdomain in 2D includes at least two vertices, since constraints at two point control the rigid body motions. In 3D we need at least three vertices with linearly independent coordinates to constrain our subdomain. Thus, three vertex constraints control the rigid body motions.

We will refer to FETI-DP algorithms in 2D and 3D using vertex constraints as the **Algorithm A**, cf. Klawonn, Widlund, and Dryja [86], where this notation was introduced.

We remark that in order to ensure a sufficient number of vertices in all cases of boundary conditions, e.g. in a structured decomposition of the unit cube into square/cubic subdomains, we may want to introduce vertices where Γ intersects $\partial\Omega_N$. This is in addition to all crosspoints, i.e. the points where the boundaries of four subdomains intersect in 2D or where eight subdomains intersect in 3D.

In fact, this strategy turns out to be successful for 2D linear elasticity, and we again get a $(1+\log(H/h))^2$ condition number bound. This can be proven using similar techniques as in Chapter 2, see also Klawonn and Widlund [84], where the proofs are given for three dimensions.

In Figure 3.1 (p. 53) and Table 3.1 (p. 55), results of numerical experiments are shown. We can see that in the 2D case the condition number grows only slowly with increasing H/h. If we plot $\sqrt{\kappa}$ using a logarithmic scale for H/h we obtain an almost straight line. This is a strong indication from the numerical computations that we indeed have a $(1+\log(H/h))^2$ bound and that it is sharp. This also means that in 2D our FETI-DP algorithm using vertex constraints is scalable with respect to the subdomain size, i.e. we are free to choose our subdomain size without any large effect on the convergence of our algorithm.

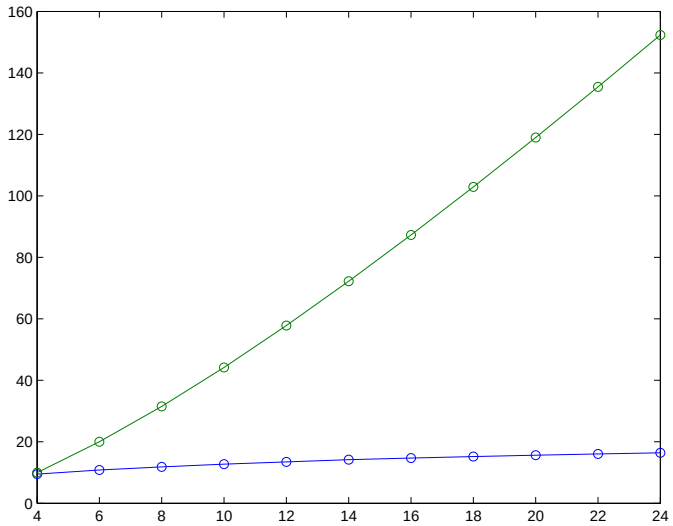

Figure 3.2
Condition number of the FETI-DP operator for linear elasticity using vertex constraints plotted versus H/h. Lower curve: 2D. Upper: 3D.

	2D (Alg. A)			3D (Alg. A)		
H/h	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
4	18	**9.70**	1.02	31	**10.51**	1.02
6	19	**11.15**	1.03	43	**20.36**	1.02
8	20	**12.24**	1.04	53	**32.01**	1.02
10	21	**13.11**	1.03	61	**44.92**	1.02
12	21	**13.84**	1.03	70	**58.81**	1.02
14	22	**14.47**	1.02	78	**73.47**	1.02
16	22	**15.02**	1.02	86	**88.80**	1.02
18	22	**15.52**	1.02	93	**104.71**	1.02
20	23	**15.97**	1.02	99	**121.10**	1.02
22	23	**16.38**	1.02	104	**137.94**	1.02
24	23	**16.75**	1.02	109	**155.19**	1.02

Table 3.1
Eigenvalues and iteration counts for FETI-DP using vertex constraints (Algorithm A) for linear elasticity in 2D and 3D, 64 subdomains, relative residual reduction of 10^{-10}.

However, as can be seen in Figure 3.2 and Table 3.1 for 3D linear elasticity, this is not the case in three space dimensions. Here, the condition number and iterations counts grow quickly with H/h. This is due to the bad properties of point constraints in three dimensions. Hence, new coarse spaces have to be constructed.

The poor performance of FETI-DP using vertex constraints has been known for some time and was first reported in Pierson [107] and Farhat, Lesoinne, and Pierson [41] where also an extension of FETI-DP to three dimensional elasticity was introduced. We will follow a somewhat different approach than the one in [41].

3.3 FETI-DP in 3D

We have to decide how to choose the primal displacement variables in 3D elasticity other than using vertex constraints. Our presentation is based on Klawonn and Rheinbach [76]. We have already discussed the choice of vertex constraints in Section 3.2. See also Farhat et al. [40], where this approach was first considered. Different approaches have been taken to construct more powerful coarse spaces. Our approach is taken from Klawonn and Widlund [84] and is motivated from the theory given therein. We will show that

an algorithm using several main ideas from Klawonn and Widlund is practicable, competitive, and robust for solving large, structured and unstructured 3D linear elasticity problems sequentially or in parallel. To our knowledge, the first parallel results for this approach were published in Klawonn and Rheinbach [76]. Note that some of the ideas, e.g. the transformation of basis for FETI-DP, have been presented already in Klawonn, Widlund and Dryja [86] and [82].

The algorithm has several desirable features and is justified from the theory. But we will also show that the algorithm is applicable when the model assumptions made for the theory do not apply. This is already the case when graph partitioners are used for the domain decomposition or when material jumps do not align with subdomain boundaries.

3.3.1 Change of Basis

In the FETI-DP methods using vertex constraints, e.g. for the 2D problems in Chapter 2, the coarse problem is built by assembling the local stiffness matrices in the primal vertices. In our 3D methods we would like to constrain certain averages to be the same across the subdomain boundaries in addition to or instead of the vertex constraints. An approach which arises naturally from the theory is to treat the average constraints in the same way as the vertex constraints after a change of basis has been carried out. Hence, we will introduce certain averages explicitly into our equation systems, and we then use subassembly to constrain these averages to be continuous across the subdomain interfaces. In Figure 3.3 a usual nodal basis in 1D and a basis including an average is shown.

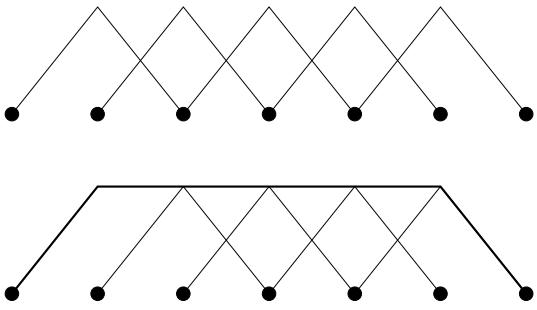

Figure 3.3
Upper: Usual nodal basis consisting of five nodal basis functions. Lower: Basis consisting of an average and four nodal basis functions.

Let us denote the transformation matrix which performs the change from

the new basis to the standard nodal basis by

$$T = \mathrm{diag}(T^{(i)}).$$

Let $K^{(i)}$ be the local subdomain stiffness matrices. The matrix

$$K = \mathrm{diag}(K^{(i)})$$

becomes

$$\overline{K} = T^T K T = \mathrm{diag}(T^{(i)T} K^{(i)} T^{(i)})$$

after transformation. We also have a new right hand side

$$T^T f = [(T^T f^{(1)})^T, \ldots, (T^T f^{(N)})^T]^T.$$

Here, we can see that the change of basis is an operation local to each subdomain.

Now, the primal variables are chosen, e.g., as averages and, possibly, as vertex unknowns. As usual, we order the primal variables last in each subdomain,

$$\overline{K}^{(i)} = \begin{bmatrix} \overline{K}^{(i)}_{BB} & \overline{K}^{(i)T}_{\Pi B} \\ \overline{K}^{(i)}_{\Pi B} & \overline{K}^{(i)}_{\Pi\Pi} \end{bmatrix},$$

arrange the local matrices in a large block matrix

$$\begin{bmatrix} \overline{K}_{BB} & \overline{K}^T_{\Pi B} \\ \overline{K}_{\Pi B} & \overline{K}_{\Pi\Pi} \end{bmatrix},$$

$$\overline{K}_{BB} = \mathrm{diag}(\overline{K}^{(i)}_{BB}), \quad \overline{K}_{\Pi B} = \mathrm{diag}(\overline{K}^{(i)}_{\Pi B}), \quad \overline{K}_{\Pi\Pi} = \mathrm{diag}(\overline{K}^{(i)}_{\Pi\Pi}),$$

and then assemble in the primal variables,

$$\widetilde{\overline{K}} = \begin{bmatrix} \overline{K}_{BB} & \widetilde{\overline{K}}^T_{\Pi B} \\ \widetilde{\overline{K}}_{\Pi B} & \widetilde{\overline{K}}_{\Pi\Pi} \end{bmatrix} = \begin{bmatrix} I_B & 0 \\ 0 & R^T_\Pi \end{bmatrix} \begin{bmatrix} \overline{K}_{BB} & \overline{K}^T_{\Pi B} \\ \overline{K}_{\Pi B} & \overline{K}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi \end{bmatrix}$$

$$\tilde{f} = \begin{bmatrix} \overline{f}_B \\ \widetilde{\overline{f}}_\Pi \end{bmatrix} = \begin{bmatrix} I_B & 0 \\ 0 & R^T_\Pi \end{bmatrix} \begin{bmatrix} \overline{f}_B \\ \overline{f}_\Pi \end{bmatrix}.$$

We now continue with the FETI-DP algorithm as described in Chapter 2. In the preconditioner we also use the transformed local stiffness matrices $\overline{K}^{(i)}$.

3.3.2 Local Lagrange Multipliers

The transformation of basis affects the sparsity of the local stiffness matrices. For our favorite coarse problem that we will describe later this will be a minor issue. But for other choices of constraints this can be a drawback particularly due to the additional effort necessary when computing the factorizations of $\overline{K}_{BB}^{(i)}$. This can be improved by a strategy of solving local saddle problems [84]. A similar technique has been used in [54] for Neumann-Neumann preconditioners to enforce zero average pressure in almost incompressible elasticity. The original formulation of BDDC [29] also solves local saddle point systems but without using global assembly of the subdomain primal variables.

We carry out our transformation of basis as described in the last section. But whenever a system

$$\overline{K}_{BB}^{(i)} \overline{u}_B^{(i)} = \overline{f}_B^{(i)}$$

has to be solved, we solve instead a constrained linear system

$$\begin{bmatrix} K^{(i)} & C^{(i)T} \\ C^{(i)} & 0 \end{bmatrix} \begin{bmatrix} u^{(i)} \\ \mu \end{bmatrix} = \begin{bmatrix} T^{(i)-T} \begin{bmatrix} \overline{f}_B^{(i)} \\ 0 \end{bmatrix} \\ 0 \end{bmatrix}.$$

We then have

$$\overline{u}_B^{(i)} = \begin{bmatrix} I_B & 0 \end{bmatrix} T^{(i)-T} u^{(i)}.$$

Here, instead of calculating the factorizations of our transformed local matrices we now solve local saddle point systems. In the factorization we keep the original stiffness matrix but constrain our averages to be zero by a Lagrange multiplier μ. The system can be solved, e.g, by LU factorization with pivoting. Using a standard technique, it can also be solved by factorizing two symmetric positive definite systems per saddle point system. Note that often it is easy to give the inverse of $T^{(i)}$ explicitly, i.e. with no computational cost. Also, the coarse matrix $\widetilde{S}_{\Pi\Pi}$, which represents the FETI-DP coarse problem, always remains symmetric positive definite as in the case when only vertex constraints are used.

In the computations in this thesis we have used the change of basis as described in Section 3.3.1. We will now describe our favorite coarse problem.

3.3.3 Edges

As we are interested only in 3D problems, we have a domain $\Omega \subset \mathbb{R}^3$ and decompose it into nonoverlapping subdomains $\Omega_i, i = 1, \ldots, N$. As usual,

each subdomain is the union of finite elements with matching finite element nodes on the boundaries of neighboring subdomains across the interface Γ.

The interface Γ then is the union of three different groups of open sets, namely, subdomain faces, edges, and vertices, cf. Figure 3.4. The face and

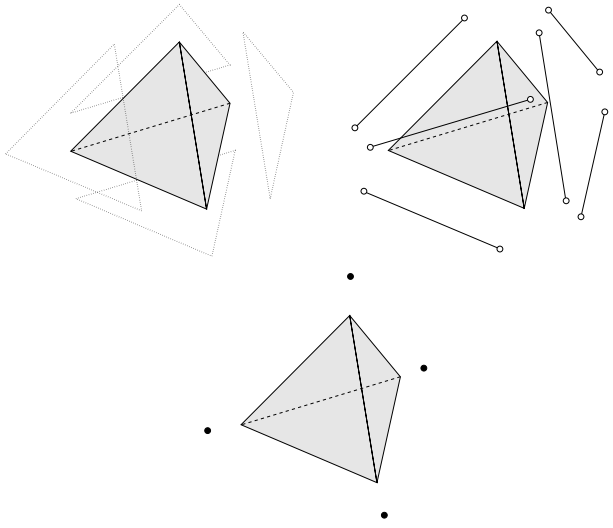

Figure 3.4
The four faces (upper left), six edges (upper right), and four vertices (lower) of a tetrahedron.

edge averages as well as the vertex degrees of freedom are potential primal variables for FETI-DP algorithms in 3D.

We need to define faces, edges, and vertices also in the case of an irregular domain decomposition as provided by a mesh partitioner, see Section 3.4.2.

Let us denote the sets of nodes of the triangulations of $\overline{\Omega}$ by Ω_h and that of $\overline{\Omega}_i$, by $\Omega_{i,h}$. Likewise we write Γ_h for the discrete interface, i.e. the triangulation nodes on the interface Γ. We denote faces, edges, and vertices in the sense of Figure 3.4 by the letters $\mathcal{F}, \mathcal{E}$, and $\mathcal{V}$, respectively.

We then consider the nodal graph

$$G_{\tau_h} = (V, E)$$

associated with a triangulation τ_h. It is defined as follows: Nodes of τ_h are nodes $v \in V$ in G_{τ_h}. Two nodes $v \in V$ are connected by an edge $e \in E$ if they share a finite element, and they are *direct neighbors* within this element.

In tetrahedral linear finite elements all nodes of an element are direct neighbors. This is not true for a hexahedral finite element, see Figure 3.5, or

for any other 3D finite element. Nevertheless, it is still simple and computationally inexpensive to build G_{τ_h} from any finite element mesh τ_h.

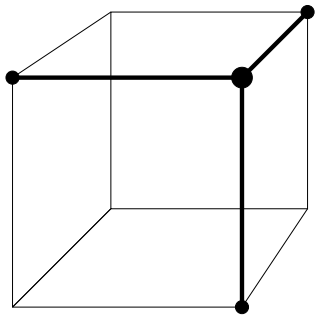

Figure 3.5: *Direct neighbors of a node in a hexahedral finite element.*

For any nodal point $x \in \Omega_h$, we also define the index set

$$\mathcal{N}_x := \big\{ i \in \{1, \ldots, N\} : x \in \Omega_{i,h} \big\},$$

i.e. $\mathcal{N}_x$ is the set of indices of all subdomains having x in the closure of the subdomain. For a node x we define the multiplicity of the node as $|N_x|$.

Sometimes it is helpful to regard nodes on the Neumann boundary $\partial\Omega_N$ as belonging to $\overline{\Omega^C} := \overline{\mathbf{R}^3 \setminus \Omega}$, in addition to the subdomains in the sense of N_x. This increases the multiplicity of nodes on $\partial\Omega_N$ by one. We then have

$$\mathcal{N}_x^C := \big\{ i \in \{1, \ldots, N\} \cup \{N^C\} : x \in (\Omega_{i,h} \setminus \partial\Omega_{D,h}) \big\},$$

where N^C is the index associated with $\overline{\Omega^C}$.

We now partition all nodes of G_{τ_h} into connectivity components with the same index set $\mathcal{N}_x$.

Definition 3.3.1 We consider

- connectivity components with $|N_x| = 1$ as the interiors of subdomains,
- connectivity components with $|N_x| = 2$ as faces,
- connectivity components with $|N_x| \geq 3$ as edges if they contain more than one node,
- otherwise we call this node a vertex.

	Vertices	Edges (Alg. D_E)	Faces
N_x^C	$(n+1)^3 - 12(n-2) - 8$	$3n(n-1)^2 + 12n(n-1)$	$3(n-1)n^2$
N_x	$(n-1)^3$	$3n(n-1)^2$	$3(n-1)n^2$

Table 3.2
Number of vertices, edges, and faces for a cubic Ω decomposed into $N = n^3$ cubic subdomains as given by Definition 3.3.1 (p. 60).

For flat structures and some thin geometries it is better to use N_x^C instead of N_x in order to define a sufficient number of edges.

In the case of a decomposition into regular substructures, e.g. cubes or tetrahedra, our definition of faces, edges, and vertices coincides with our basic geometric intuition, see Figures 3.4 (p. 59) and 3.7 (p. 63). On the other hand, for decompositions generated by a graph partitioner, the situation can be more complex. We can, e.g., have several edges with the same index set $\mathcal{N}_x$ or also an edge and a vertex with the same $\mathcal{N}_x$. In an implementation, the faces, edges, and vertices can be computed by usual graph traversing algorithms, e.g. depth first search. In practice, we can also have situations when there are not enough edges and potential edge constraints for some subdomains. In these cases we can switch to using N_x^C, which also can be done locally, i.e. for a single pair of subdomains. We will also increase the number of edges in the irregular case by splitting edges into several edges whenever needed. It is our experience that in decompositions coming from automatic graph partitioners almost all edges ($> 99\%$) have a multiplicity of three.

Let us note that sometimes the term edge is used differently in the literature; cf., e.g., [41, 107]. Also, instead of vertices, the term corners is often used although usually with a different meaning than vertices in our terminology; cf. [41, 107, 88]. A simple choice is to define corners in three dimensions as nodes which either belong to more than three subdomains, see Pierson [107, p.24], or to more than four subdomains, see Lesoinne [88]. A more elaborate algorithmic choice, which can reduce the number of corners, is given in [88]. For completeness, we also give the definition of edges in three dimensions as used in [41, 107]; see Pierson [108]. There, an "edge" between two subdomains is the set of interface nodes, excluding corners, which is common to the boundaries of both subdomains. As pairs of subdomains are considered, one such edge is defined for each two subdomains that share nodes. To illustrate the subtle difference between this and our definition, we consider a structured decomposition of the unit cube into cubic subdomains.

For two different situations of interior neighboring subdomains, we illustrate the different definitions in Figure 3.6 (p. 62) and Figure 3.7 (p. 63). An "edge" in Figure 3.6 can be, e.g., either an edge as in Definition 3.3.1 or the union of, in our terminology, a face with the edges belonging to its closure; cf. also Figure 3.7. It is worth to point out that this definition generates as many "edges" as the sum of edges and faces of our Definition 3.3.1. In our algorithm the coarse degrees of freedom on the edge are shared by four subdomains. This will reduce the size of our coarse problem.

Let us note that for the algorithms in this chapter, we solely focus on the use of constraints over edges. For homogeneous problems no vertices or faces are needed.

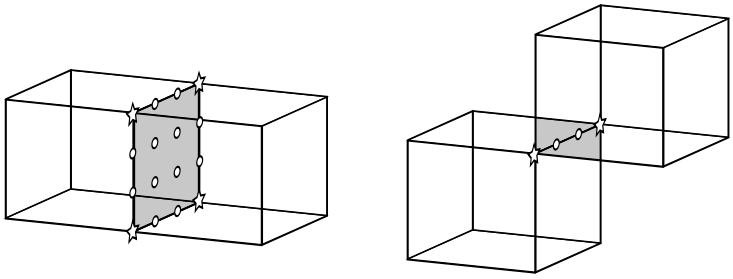

Figure 3.6
Illustration of the definition of edges as given in [41, 108, 107]. When the nodes, denoted by stars, are selected as the corners, the set of edge nodes is given by those nodes denoted by circles.

3.3.4 Change of Basis for Edge Constraints

We now describe in detail our approach using an explicit change of basis for edge constraints. We use the transformation of basis to change between the standard finite element nodal basis and a basis where we have introduced edge averages explicitly into the system. We will then choose the edge averages over the three components as the primal variables.

The transformation matrix T_E performs the desired change of basis from the new basis to the original nodal basis since we would like to iterate in the new finite element space. Denoting the edge unknowns in the new basis by $\overline{u}_E$, we have

$$u_E = T_E \overline{u}_E.$$

Such a transformation matrix T_E can be constructed separately for each edge with three primal edge constraints. Ordering the three edge averages

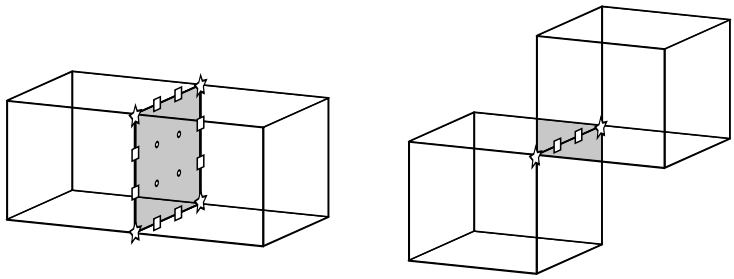

Figure 3.7
Illustration of Definition 3.3.1 of vertices, edges, and faces. Vertices are denoted by stars, edge nodes by squares, and face nodes by small circles. Left: The intersection of the closure of the two subdomains consists of one face, four edges, and four vertices. Right: The intersection of the closure of the two subdomains consists of one edge and two vertices.

last, a possible choice of T_E is

$$T_E = \begin{bmatrix} I_3 & & 0 & I_3 \\ & \ddots & & \vdots \\ 0 & & I_3 & I_3 \\ -I_3 & \cdots & -I_3 & I_3 \end{bmatrix}, \tag{3.8}$$

where I_3 is the 3×3 identity matrix. We denote the resulting transformation which operates on all relevant edges of $\partial\Omega_i$ by $T_E^{(i)}$. The transformation for all variables of one subdomain Ω_i is then of the form

$$T^{(i)} = \begin{bmatrix} I_I & 0 & 0 \\ 0 & I_{\overline{\Gamma}} & 0 \\ 0 & 0 & T_E^{(i)} \end{bmatrix}.$$

Here, I_I and $I_{\overline{\Gamma}}$ are identity matrices, and we assume that the variables are ordered interior variables first (I), then interface variables not related to the primal edges ($\overline{\Gamma}$), and then the variables on the primal edges (E). A typical vector of nodal unknowns is of the form

$$[u_I^{(i)T}, u_{\overline{\Gamma}}^{(i)T}, u_E^{(i)T}]^T.$$

The subdomain transformation $T_E^{(i)}$ is the direct sum of the relevant transformation matrices associated with the primal edges of that subdomain; $T_E^{(i)}$ is a block-diagonal matrix, where each block represents the transformation of a component of a primal edge.

Decomposing the subdomain stiffness matrices $K^{(i)}$ in the same manner, we obtain

$$K^{(i)} = \left[\begin{array}{cc|c} K_{II}^{(i)} & K_{I\overline{\Gamma}}^{(i)} & K_{IE}^{(i)} \\ K_{\overline{\Gamma}I}^{(i)} & K_{\overline{\Gamma}\overline{\Gamma}}^{(i)} & K_{\overline{\Gamma}E}^{(i)} \\ \hline K_{EI}^{(i)} & K_{E\overline{\Gamma}}^{(i)} & K_{EE}^{(i)} \end{array}\right].$$

Using the transformation $u^{(i)} = T^{(i)}\overline{u}^{(i)}$, we get

$$T^{(i)T}K^{(i)}T^{(i)} = \left[\begin{array}{cc|c} K_{II}^{(i)} & K_{I\overline{\Gamma}}^{(i)} & K_{IE}^{(i)}T_E^{(i)} \\ K_{\overline{\Gamma}I}^{(i)} & K_{\overline{\Gamma}\overline{\Gamma}}^{(i)} & K_{\overline{\Gamma}E}^{(i)}T_E^{(i)} \\ \hline T_E^{(i)T}K_{EI}^{(i)} & T_E^{(i)T}K_{E\overline{\Gamma}}^{(i)} & T_E^{(i)T}K_{EE}^{(i)}T_E^{(i)} \end{array}\right], \tag{3.9}$$

where the upper left 2×2 block matrix is not affected by the basis transformation. The primal variables in the new basis now consist of averages. We order them last to get,

$$T^{(i)T}K^{(i)}T^{(i)} = \left[\begin{array}{ccc} K_{II}^{(i)} & \overline{K}_{\Delta I}^{(i)T} & \overline{K}_{\Pi I}^{(i)T} \\ \overline{K}_{\Delta I}^{(i)} & \overline{K}_{\Delta\Delta}^{(i)} & \overline{K}_{\Pi\Delta}^{(i)T} \\ \overline{K}_{\Pi I}^{(i)} & \overline{K}_{\Pi\Delta}^{(i)} & \overline{K}_{\Pi\Pi}^{(i)} \end{array}\right].$$

Using the assembly operators $R_\Pi^{(i)T}$, $R_\Pi^{(i)}$ to assemble the primal variables and ordering them last in the partially assembled matrix, we obtain

$$\widetilde{K} := \left[\begin{array}{ccccccc} K_{II}^{(1)} & \overline{K}_{I\Delta}^{(1)} & & & & & \widetilde{K}_{\Pi I}^{(1)T} \\ \overline{K}_{\Delta I}^{(1)} & \overline{K}_{\Delta\Delta}^{(1)} & & & & & \widetilde{K}_{\Pi\Delta}^{(1)T} \\ & & \ddots & & & & \vdots \\ & & & K_{II}^{(N)} & \overline{K}_{I\Delta}^{(N)} & & \widetilde{K}_{\Pi I}^{(N)T} \\ & & & \overline{K}_{\Delta I}^{(N)} & \overline{K}_{\Delta\Delta}^{(N)} & & \widetilde{K}_{\Pi\Delta}^{(N)T} \\ \widetilde{K}_{\Pi I}^{(1)} & \widetilde{K}_{\Pi\Delta}^{(1)} & \cdots & \widetilde{K}_{\Pi I}^{(N)} & \widetilde{K}_{\Pi\Delta}^{(N)} & & \widetilde{K}_{\Pi\Pi} \end{array}\right] =: \left[\begin{array}{cc} \overline{K}_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{array}\right].$$

As we have pointed out earlier, the transformation of basis changes the sparsity pattern of the transformed matrices $T^{(i)T}K^{(i)}T^{(i)}$ compared to that of the original local stiffness matrices $K^{(i)}$. Only the matrix blocks related to the edge degrees of freedom, and those coupling with them, are affected; cf. (3.9). We note that the set of edge degrees of freedom is only a small subset of the overall set of degrees of freedom. Thus, the transformation of basis only has a minor effect on the sparsity pattern.

To illustrate the change of sparsity of our local, transformed matrices $T^{(i)T}K^{(i)}T^{(i)}$ using the transformation (3.8), we investigated the increase of nonzero entries in $T^{(i)T}K^{(i)}T^{(i)}$ compared to $K^{(i)}$ for the problems in Section 3.5 (Tables 3.5 and 3.6). We found an increase between 4 and 12 percent of nonzero entries compared to the original local matrices $K^{(i)}$.

In our FETI-DP algorithm we always assume that we have performed an appropriate change of basis. If there is no danger of confusion we will drop the overline notation which indicates the dual displacement variables in the transformed basis.

Using the transformation of basis, we again obtain

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

We note that, after the change of basis has been carried out, we can always use the same implementation as when using vertex constraints; the algorithmic description in Section 2.2 does not depend on a specific choice of primal and dual variables. Note that the local problems as well as the Schur complement $\tilde{S}_{\Pi\Pi}$ remain symmetric positive definite.

3.3.5 Choice of Coarse Problem

From an algebraic point of view, we need to choose sufficiently many constraints to make $\widetilde{K}$ invertible. For this, we use our edge constraints on primal faces. If we have six linearly independant edge constraints for a face we call this a *fully primal face* [84]. If a face is fully primal, all six rigid body motions are controlled on this face and the two subdomains sharing this face can only move as a union. We have seen that using vertices (alone) will not lead to a good condition number bound, see Section 3.2 and Figure 3.2 (p. 54). This is the reason why we drop vertices completely from our coarse space, except for some exotic configurations in heterogeneous elasticity, see Chapter 5.

Hence, let us define the primal face graph $G_{\mathcal{F}}$. An example can be seen in Figure 3.8.

Definition 3.3.2 *(Primal Face Graph) We regard the subdomains as the nodes $v \in V$ in the primal face graph $G_{\mathcal{F}} = (V, E)$, and two such nodes are connected by an edge $e \in E$ if the corresponding subdomains have a fully primal face in common.*

Note that by making a face fully primal, as we use edge constraints, neighboring faces are affected as well. This way additional fully primal faces can

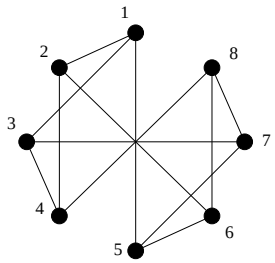

Figure 3.8
Primal face graph $G^2_{\mathcal{F}}$ for a cubic domain decomposed into $N = 2^3$ cubic subdomains if all faces are primal.

arise "automatically" from certain configurations of fully primal faces. This is due to the fact that we assemble every coarse degree of freedom from the edge averages of at least three subdomains. For the following considerations we will neglect these fully primal faces that automatically arise.

We then have invertibility of $\widetilde{K}$ as soon as $G_{\mathcal{F}}$ is a connected graph. The minimal number of fully primal faces is given for a connected subgraph of $G_{\mathcal{F}}$ with the minimal number of edges $e \in E$. This is the case for any spanning tree of $G_{\mathcal{F}}$. As we have costs of one on each edge, any spanning tree is a minimal spanning tree, and all spanning trees have a cost of $N - 1$.

However, to ensure a good bound on the condition number, we need to fulfill additional conditions.

Definition 3.3.3 *For a subdomain Ω_i we call a subdomain Ω_j a geometric neighbor if $\overline{\Omega}_i \cap \overline{\Omega}_j \neq \emptyset$. We define the length of the longest path in $G_{\mathcal{F}}$ from a subdomain to a geometric neighbor as* $\mathrm{path}(G_{\mathcal{F}})$*, i.e.*

$$\mathrm{path}(G_{\mathcal{F}}) := \max_{\overline{\Omega}_i \cap \overline{\Omega} \neq \emptyset} (l_{ij}),$$

where l_{ij} is the length of the shortest path from Ω_i to the geometric neighbor Ω_j.

In the case of homogeneous linear elasticity the following condition assures a good bound on the condition number.

Condition 3.3.1 *(Path) For all neighbors Ω_j of a subdomain Ω_i we have a path of uniformly bounded length L from Ω_i to Ω_j, i.e.*

$$\mathrm{path}(G_{\mathcal{F}}) \leq L.$$

In this case we have what Klawonn and Widlund call an acceptable set of primal constraints [84, Definition 6].

Then, the following condition number estimate for FETI-DP algorithms using edge averages on selected edges as primal constraints can be deduced from Klawonn and Widlund [84, Theorem 1].

Theorem 3.3.1 *The condition number satisfies*

$$\kappa(M^{-1}F) \leq C\,(1+\log(H/h))^2,$$

where H denotes the subdomain diameter and h the finite element mesh size. The positive constant C is independent of h and H.

For homogeneous materials or problems with discontinuities of the material coefficients which are not very large, it follows that primal vertices are not needed to obtain a good condition number bound.

We can easily construct a (minimal) spanning tree where the length of the path in the sense of Condition (3.3.1) is $O(N)$. To see this, let us consider the case of a structured decomposition of a cube into $N = n^3$ cubic subdomains. It is easy to create a path of length $n^3 - 1$ from Ω_1 to Ω_{n^3}. For path($G_{\mathcal{F}}$) we need to consider paths between geometrically neighboring subdomains, only. Hence, we consider $(n+1)^3$ subdomains. From a subdomain Ω_i on the surface of our cubic domain, i.e. $\overline{\Omega}_i \cap \partial\Omega \neq \emptyset$, we can construct a path entering an n^3 subcube, traversing this, and then returning to a geometric neighbor of Ω_i. We then take care of all remaining subdomains to create a spanning tree.

Note that we slightly misuse the Laundau notation since two of the symbols to make correct use of the Laundau notation, i.e. Ω and Θ, are already in use.

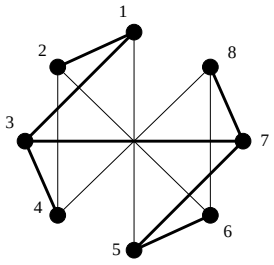

Figure 3.9: *A (minimal) spanning tree for the graph in Figure 3.8.*

Can we construct spanning trees with better bounds on the path length? Let us assume that we have a spanning tree of $G_{\mathcal{F}}$ which satisfies Condition 3.3.1. As we have a tree, by removing any edge it is separated into two

connected components. We can find an edge that separates the tree into two components of similar size $O(n^3)$. Let us denote this edge by e, the subdomains which it connects by Ω_1 and Ω_2, and the components by S_1 and S_2, where $\Omega_1 \in S_1$, $\Omega_2 \in S_2$. The best bounds on the size of good separators for planar graphs [94] give us that components S_1 and S_2 have an interface of $O(n)$ subdomain faces. Note that the bound in [94] is an optimal bound. Also note that for graphs with bounded degree we can construct partitions with small edge cuts from small separators and vice versa by considering the nodes adjacent to the edges. For graphs from 3D finite element meshes we can indeed expect $O(n^2)$ subdomain faces, see [101]. If the number of faces of a subdomain is bounded, we can also find $O(n)$ pairs $(\Omega_{1,i}, \Omega_{2,i})$, where $\Omega_{1,i} \in S_1$, $\Omega_{2,i} \in S_2$, and $\Omega_{1,i}$, $\Omega_{2,i}$ are geometric neighbors. All paths connecting such pairs have to use the edge e and are, from our assumption, bounded in length by a constant L. This means that all such pairs must reside near Ω_1 and Ω_2, i.e. they must reside in a sphere of radius $O(L)$ around Ω_1 and Ω_2. We can use, e.g., the Manhattan distance to define this sphere. If we assume quasi-uniformity of the subdomains then there is not enough room to fit $O(n)$ subdomains into this sphere.

In fact, assuming the more pessimistic bound of $O(n^2)$ pairs, we can deduce a lower bound from comparing the number of subdomains with the volume of the sphere which is $O(n^{2/3}) = O(N^{2/9})$ for the length of the path. This means that we cannot fulfill Condition 3.3.1 without introducing cycles into $G_{\mathcal{F}}$. In fact, for any such decomposition of $G_{\mathcal{F}}$ into two equilibrated subgraphs, we must require that $O(n^2)$ edges are cut if we want to guarantee a constant path length. Here we have assumed the more pessimistic bound for 3D graphs.

From these considerations, we are can now use local heuristics that may induce cycles, e.g., one can calculate small spanning trees in neighborhoods with a fixed diameter. This can be implemented as a linear time algorithm, and a maximum path length can be guaranteed. Note that for a given graph we can verify Condition 3.3.1 by an all pair shortest path (apsp) algorithm.

We have seen that the choice of $N-1$ fully primal faces is not sufficient. On the other hand, our Condition 3.3.1 is clearly satisfied if all faces are made primal, i.e. if $L = 1$. In the structured case, see Table 3.2 (p. 61), this corresponds to $3N$ fully primal faces.

In our experience, the condition number can deteriorate substantially with the length of the path, even if it is bounded by a constant. From the proof in Chapter 2 for scalar problems, it becomes clear that increasing the path will increase the condition number bound accordingly. Let us assume that all subdomains contribute the same value to the estimate and that the bound is $2C$ for a path length of one. Then increasing the path length by

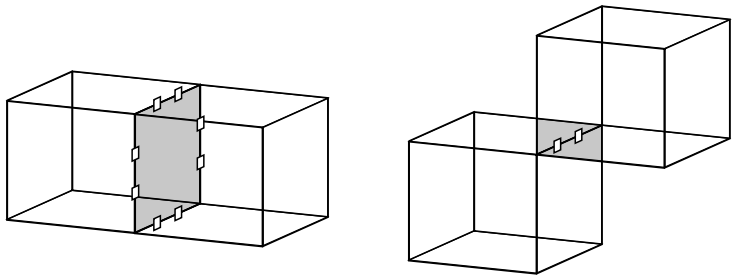

Figure 3.10: *Algorithm D_E only uses edge averages.*

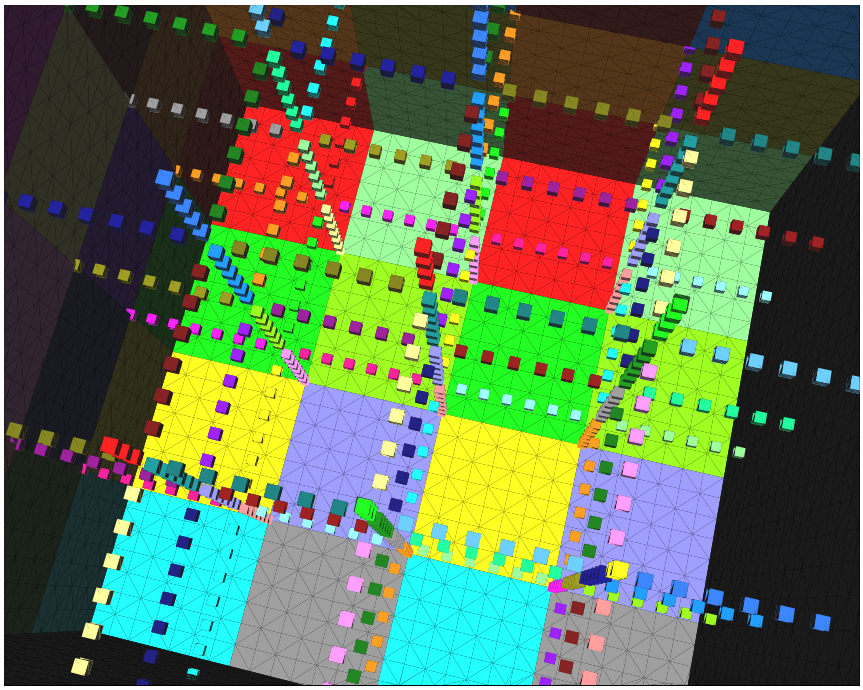

Figure 3.11
Edge nodes (small cubes) are shown for a decomposition of the unit cube decomposed into cubic subdomains, cf. Figure 3.10. Nodes with the same color belong to the same edge. The subdomains are invisible, $\partial\Omega$ is seen in the background from the interior.

one will change the bound from $2C$ to $3C$. For our computations we will therefore prefer the coarse space where all faces are fully primal, i.e. the path length L is one. The situation can be different in the heterogeneous setting as we see in the experiments in Section 5.3.

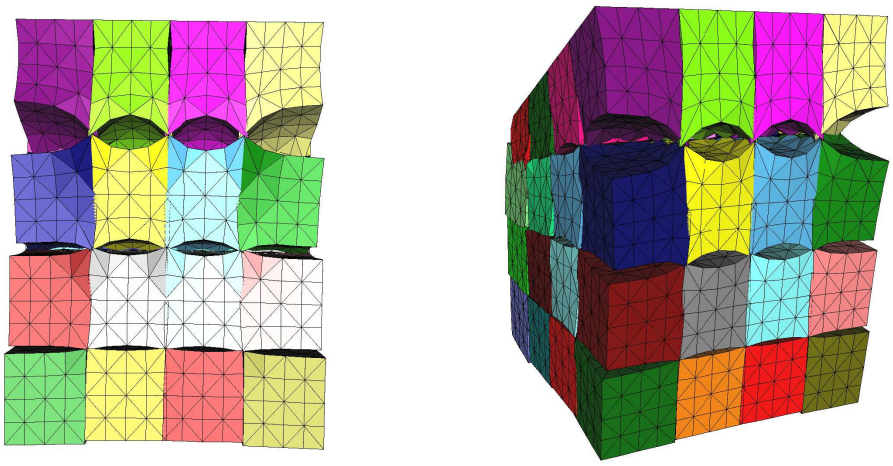

Figure 3.12
Algorithm D_E: Initial solution before the start of the conjugate gradient iterations. We have continuity of edge averages at all edges given by Definition 3.3.1 using N_x. The number of edges corresponds to the left image in Figure 3.16 (p. 80).

3.3.6 Algorithm D_E

In order to define all faces as fully primal we can use the edge averages on all edges defined by Definition 3.3.1. In some special cases it may be necessary to introduce additional constraints by splitting an edge into two or several edges. We thus introduce a new notation and denote the FETI-DP algorithm using edge averages on all edges, and using these as the only primal constraints, by **Algorithm D_E**. We note again that, algorithmically, the edge average (Algorithm D_E) constraints can be treated, by using the transformation of basis, in the same way as the vertex constraints (Algorithm A).

We note that in Chapter 5 a variant of Algorithm D_E is considered where additionally first order moments on selected edges and possibly selected primal vertices are used, see also Klawonn and Rheinbach [77]. In this case, a condition number estimate is shown which is also independent of discontinuities of the material coefficients across the interface [84].

In Section 3.2, we discussed the need for a coarse space appropriate for 3D problems. In Table 3.3 (p. 72) and Figure 3.13 (p. 71) we now compare the condition number for Algorithm D_E with the values given in Table 3.1 (p. 55) for Algorithm A. Clearly, Algorithm D_E shows the desired asymptotic

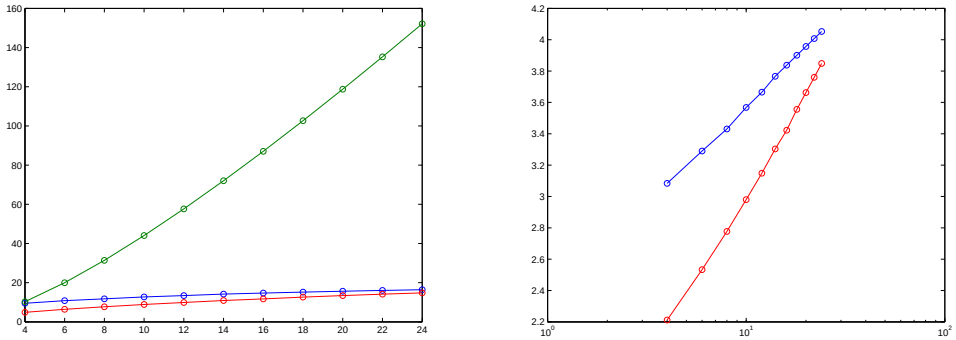

Figure 3.13

Graphs from Figure 3.2 (p. 54), now including Algorithm D_E. Condition number of the FETI-DP operator for linear elasticity plotted versus H/h. Left: 3D using vertex constraints (upper curve), 2D using vertex constraints (middle curve), 3D using edge averages, i.e. Algorithm D_E (lower curve). Right: Semilogarithmic plot of $\sqrt{\kappa}$ versus H/h for 2D using vertex constraints (upper) and for 3D using edge averages, i.e. Algorithm D_E (lower).

behavior and the condition number even stays below the values obtained for Algorithm A in 2D.

3.3.7 Algorithm D_F

Although this case is not addressed in the theory in [76] the rigid body modes of a face can also be controlled by introducing three average constraints and three first order moment constraints on each face. In this case no vertex constraints and no edge constraints are introduced. This algorithm also shows scalability with respect to the number of subdomains, see Table 3.4 (p. 72) but the coarse problem is larger, see Table 3.2 (p. 61). We enforce three average constraints per edge in Algorithm D_E but we have six constraints per face in Algorithm D_F. A similar condition number bound as in Theorem 2.4.2 can be proven for Algorithm D_F by analogous techniques as in [84]; see also Kim [71] for a theoretical treatment of a similar algorithm in the case of mortar element discretizations.

3.3.8 Optional Lagrange Multipliers

There is a different and earlier approach to implement average constraints which has proven very successful in computational practice. It uses optional

	2D (Alg. A)			**3D (Alg. A)**			**3D (Alg. $\mathbf{D_E}$)**		
H/h	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
4	18	**9.70**	1.02	31	**10.51**	1.02	21	**4.99**	1.02
6	19	**11.15**	1.03	43	**20.36**	1.02	24	**6.48**	1.01
8	20	**12.24**	1.04	53	**32.01**	1.02	27	**7.79**	1.01
10	21	**13.11**	1.03	61	**44.92**	1.02	29	**8.97**	1.01
12	21	**13.84**	1.03	70	**58.81**	1.02	30	**10.01**	1.01
14	22	**14.47**	1.02	78	**73.47**	1.02	32	**11.02**	1.01
16	22	**15.02**	1.02	86	**88.80**	1.02	33	**11.83**	1.01
18	22	**15.52**	1.02	93	**104.71**	1.02	35	**12.77**	1.01
20	23	**15.97**	1.02	99	**121.10**	1.02	36	**13.55**	1.01
22	23	**16.38**	1.02	104	**137.94**	1.02	37	**14.28**	1.01
24	23	**16.75**	1.02	109	**155.19**	1.02	38	**14.96**	1.01

Table 3.3

Numbers from Table 3.1, p. 3.1, now including Algorithm D_E. Eigenvalues and iteration counts for FETI-DP using vertex constraints (Algorithm A) for linear elasticity in 2D and 3D, and using edge constraints in 3D (Algorithm D_E), 64 subdomains. The stopping criterion is a relative residual reduction of 10^{-10}.

	Algorithm $\mathbf{D_F}$								
N	**8**	**27**	**64**	**125**	**216**	**343**	**512**	**729**	**1000**
It.	16	19	20	21	21	21	21	21	21
κ	3.52	3.76	3.88	3.93	3.96	3.97	3.98	3.99	3.99

Table 3.4

Condition number for Algorithm D_F, 3D linear elasticity, $H/h = 5$.

Lagrange multipliers which form a part of the global coarse problem, cf. Farhat, Lesoinne and Pierson [41].

In [41], additional, optional constraints of the form

$$Q_B u_B = 0$$

and Lagrange multipliers μ were introduced to enforce continuity of certain averages of FETI-DP iterates. The FETI-DP master system then takes the

form

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & Q_B^T & B_B^T \\ \widetilde{K}_{\Pi B}^T & \widetilde{K}_{\Pi\Pi} & 0 & 0 \\ Q_B & 0 & 0 & 0 \\ B_B & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \mu \\ \lambda \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \\ 0 \end{bmatrix}. \tag{3.10}$$

This system is solved iteratively after eliminating u_B and $[\tilde{u}_\Pi^T, \mu^T]^T$. The FETI-DP system then is

$$F\lambda = d,$$

where

$$F := -B_B K_{BB}^{-1} B_B^T + \begin{bmatrix} -\widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T \\ -Q_B^T K_{BB}^{-1} B_B^T \end{bmatrix}^T (\widetilde{S}_{\Pi\Pi}^*)^{-1} \begin{bmatrix} -\widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T \\ -Q_B^T K_{BB}^{-1} B_B^T \end{bmatrix}, \tag{3.11}$$

$$\widetilde{S}_{\Pi\Pi}^* := \begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B} K_{BB}^{-1} Q_B^T \\ -Q_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T & -Q_B K_{BB}^{-1} Q_B^T \end{bmatrix}.$$

The Lagrange multipliers are optional in the sense that they are a linear combination of the continuity constraints $B_B u_B = 0$, i.e. the condition

$$Q_B u_B = q^T B_B u_B = 0$$

is enforced in every iteration. Nevertheless, the invertibility of $\widetilde{K}$ still has to be guaranteed by using vertex constraints. They are necessary because the elimination of $[u_B^T, \tilde{u}_\Pi^T]^T$ is done in a first step even though, in principle, the upper 3×3 block may be invertible.

This approach, using average constraints different from the ones used in this work, see Section 3.3.3, has been used very successfully in 3D structural mechanics simulations [41] and for large scale parallel computations [14]. It is routinely in use, e.g., in an implicit structural dynamics code [110], see Pierson et al. [109].

Here, the matrix $\widetilde{S}_{\Pi\Pi}^*$ is not symmetric positive definite anymore. The coarse problem can still be solved in two steps using Cholesky decompositions.

In an earlier FETI-DP implementation [78] we have also used this approach to implement optional edge constraints, where edges are meant in the sense of Section 3.3.3, Definition 3.3.1. In this case, the edges are in addition to vertex constraints, which are still needed here, and without a transformation of basis. Moreover, for the edges we have to use several (multiplicity of the edge minus one) optional Lagrange multipliers where we have only one coarse degree of freedom in the transformation of basis approach.

A problem that we have observed, when using optional Lagrange multipliers, is instability when solutions of high accuracy have to be computed. This occurred in some cases of ill conditioned problems even if they were well conditioned after preconditioning.

In Figure 3.14 we depict the convergence history of a two dimensional almost incompressible elasticity problem, cf. Section 6, where we chose the Poisson ratio $\nu = 0.4999$. The FETI-DP algorithm using only vertex constraints (Alg. A) reaches higher accuracy. It converges slowly but remains stable until stagnation. In a next step, we introduce optional Lagrange multipliers, in addition to our vertex constraints, to speed up the algorithm. We observe faster convergence in the beginning but as a higher accuracy is approached, the convergence of the CG method becomes unstable. The reason for this is a very small artificial eigenvalue of the order of 10^{-12}, which spoils the condition number and the convergence of the CG method. Note that we have used the same FETI-DP implementation, except for the optional Lagrange multipliers, and thus also the same implementation of conjugate gradients. The small artifical eigenvalue can be explained as follows.

Let us define $Q := [Q_B; 0]$. In the algorithm, we can formally split the elimination of u_B and μ into two steps. The Schur complement

$$\begin{bmatrix} -Q\widetilde{K}^{-1}Q^T & -Q\widetilde{K}^{-1}B^T \\ -B\widetilde{K}^{-1}Q^T & -B\widetilde{K}^{-1}B^T \end{bmatrix} = -\begin{bmatrix} q^T \\ I \end{bmatrix} B\widetilde{K}^{-1}B^T \begin{bmatrix} q & I \end{bmatrix}, \quad (3.12)$$

has a large null space since we have incorporated redundant constraints. The upper left block $Q\widetilde{K}^{-1}Q^T$ is regular. From roundoff error in the elimination of the upper left block, very small nonzero eigenvalues can appear. This corresponds to (3.10) becoming inadmissible.

When using a transformation of basis we have not observed this kind of instability, see Chapter 6 and Figure 3.22.

Since vertex constraints are still needed when the FETI-DP coarse space is augmented using optional Lagrange multipliers, there is an interest to reduce the number of the vertices. An algorithm for the choice of vertices was introduced in Lesoinne [88]. The idea is to choose a comparably small number of vertices to guarantee invertibility of $\widetilde{K}$ and otherwise enhance the coarse problem by averages using optional Lagrange multipliers.

3.4 Parallel Implementation

In scientific computing, the performance penalty of object oriented languages is still an issue when implementing low level functions. In these cases Fortran

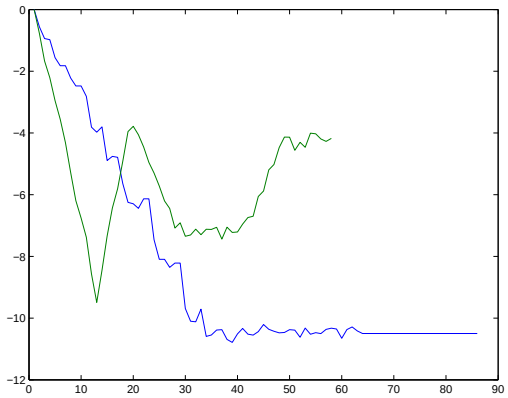

Figure 3.14
Convergence history of the relative true dual residuals. The algorithm using only vertex constraints stagnates at a lower level and remains stable although converging slower. The method using optional Lagrange multipliers in addition to the vertex constraints converges faster at first but then it becomes unstable. We use the same implementation of CG in both cases.

(lapack and blas) or even assembly language (libgoto by Kazushige Goto) are still preferred. Yet, the classic blas and lapack libraries are prime examples for successful abstract interfaces. Today, object oriented languages are routinely in use at the abstract level of algorithm formulation.

We have implemented a parallel FETI-DP algorithm in C/C++ using edge constraints as described in Section 3.3.3 and a transformation of basis as described in Section 3.3.1 and 3.3.4. We have made heavy use of external libraries to provide underlying functionality. We always used optimized libraries for our specific hardware whenever available, see below.

3.4.1 Distributed Computing and Message Passing

The primary target architecture for domain decomposition methods and even more so for Schur complement methods are parallel distributed memory machines. Distributed memory computers consist of more or less independent computers (nodes) which are connected by one or several fast networks. On each node a program is executed which performs calculations on local data and exchanges information with other nodes over the network. Despite the use of high speed networks the communication is slower by orders of magnitude in bandwidth as well as in latency compared to the access to the local

memory of the node.

Although domain decomposition algorithms also perform well on serial machines or computing cluster of modest size they belong to the class of algorithms that particularly excel on massively parallel machines. The amount of communication in domain decomposition algorithms often grows only as fast as the surface of the subdomains, except for the communication related to the small coarse problem, and most of the communications is of the nearest-neighbor type as opposed to all-to-all communication. Moreover, the cost of investing considerable effort in subdomain local solves, e.g. by the use of a direct sparse solvers in Schur complement methods, is amortized if communication is reduced.

Today's standard for interprocessor communication on distributed memory machines is the Message Passing Interface (MPI), see, e.g., [63]; a popular implementation, which we also use on our own Opteron computing cluster, is MPICH [59, 62, 60] by the Argonne National Laboratory. Often vendors supply their own optimized implementation of MPI with their machine. MPI performs equally well on shared memory computers making MPI applications portable to a wide variety of architectures. Our FETI-DP implementation uses MPI for parallelization but can also efficiently be used on a single processor system.

3.4.2 Graph Partitioning

Domain decomposition methods rely on graph partitioners to create the decomposition of the computational domain into subdomains. Generally, the dual graph is partitioned, i.e. the graph defined by the face-to-face connections of finite elements. This is opposed to direct methods, which rely on the sparsity pattern of the sparse matrix. The latter case corresponds to the partitioning of the primal graph.

Unfortunately, graph partitioning is an NP complete problem. Until recently, no polynomial time approximation algorithm was known even for the Minimum Bisection Problem, i.e. the task to partition a graph into two equilibrated parts while cutting the minimal number of edges. However, many heuristics have been developed, implemented, and successfully applied to a large variety of problems. Examples for graph partitioning software are Metis [69], ParMetis [70], and Chaco [65]. Graph partitioning can be based on very different approaches, e.g. ParMetis uses multilevel k-way partitioning with Kernighan-Lin refinement. It is based on the observation that a good partitioning for a fine graph can be found by calculating a good partitioning for a coarser graph. On the other hand, Chaco provides spectral partitioning schemes, which are based on a connection of good partitionings

with the low eigenvalues of a certain matrix related to the graph. Chaco also provides a multilevel method where only the smallest graph is partitioned by the spectral method.

Note that the objective functions minimized in graph partitioning are not completely congruent with the goals in domain decomposition. We are interested in equilibrating the computational work among the processors as well as equilibrating the communication, and we are interested in a minimal total interface in order to minimize the total communication. However, depending on the computational domain and the discretization, the computational work may not always directly correspond to the number of nodes. But even more importantly, for iterative methods we are interested in shape regularity of the subdomains since subdomains with bad aspect ratios will lead to higher condition numbers and thus more iterations. This objective is not explicitly addressed when using standard graph partitioning. Still, the computational practice shows that graph partitioners generally provide good domain decompositions also in respect to subdomain aspect ratios.

For our computations, we have used both, Chaco as well as ParMetis. However, for all examples in this work we used ParMetis to compute the domain decomposition.

3.4.3 PETSc

The PETSc (Portable Extensible Toolkit for Scientific Computing) package [9, 8, 10] is a library of data types and routines, accessible through abstract interfaces, meant to provide the building blocks for portable, parallel applications in scientific computing. It is written in C but has language bindings to all major languages. The library PETSc provides efficient implementations of sequential, as well as parallel, matrix and vector data structures and also solver objects like Krylov subspace methods. PETSc is our main tool in implementing the parallel FETI-DP algorithm and is developed at the Argonne National Laboratory. The parallel computations in this work were carried out using PETSc on top of MPI.

3.4.4 Parallel FETI-DP

The primal sources of concurrency in the FETI-DP method are the parallel factorizations of the subdomain matrices $K_{BB}^{(i)}$ and $K_{II}^{(i)}$ in the setup phase and the corresponding parallel forward-backward substitutions in the iteration phase.

These factorizations are completely independent and do not require any communication among processors. After the factorizations have been per-

formed, the coarse matrix $\widetilde{S}_{\Pi\Pi} = \widetilde{K}_{\Pi\Pi} - \widetilde{K}_{\Pi B} K_{BB}^{-1} \widetilde{K}_{\Pi B}^T$ must be computed and assembled. This is done by summing the contributions from all processors,

$$\widetilde{S}_{\Pi\Pi} = R_\Pi \left(\mathrm{diag}_{i=1}^N \left(K_{\Pi\Pi}^{(i)} - K_{\Pi B}^{(i)} (K_{BB}^{(i)})^{-1} K_{\Pi B}^{(i)T} \right) \right) R_\Pi^T,$$

into a distributed matrix. Depending on the size of the coarse problem, we can either copy the matrix to all processors as soon as it is assembled and then factor it on each, or we can use a parallel direct solver on it. In all FETI-DP experiments in this chapter we have chosen to broadcast the coarse matrix to all processors and factor it there. This means that the coarse problem is solved sequentially. This is an efficient choice for coarse problems of small or medium size. For large coarse problems a parallel exact or inexact solution of the coarse problem is necessary, see Chapter 4. In the standard FETI-DP formulation we cannot use an inexact solver. In Chapter 4 we therefore propose a new, modified algorithm. In the present chapter, for standard FETI-DP, we can use a sequential or parallel iterative coarse solver, but we must iterate until machine precision.

We have organized the FETI-DP solver around a FETI-DP matrix class which stores the distributed system matrix. Similarly, we have a distributed FETI-DP preconditioner class. Both are derived from abstract matrix and preconditioner classes and can be used with PETSc's Krylov subspace methods, see Figure 3.15. The FETI-DP matrix object can hold several subdomains per processor but subdomains cannot be shared by processors.

Most of the communication takes place in the iteration phase as in every iteration we require two updates from every subdomain (one for the system matrix and one for the preconditioner) and one from the coarse problem. After we have reached the desired accuracy we calculate our solutions $u^{(i)}$ from the Lagrange multiplier λ.

On the Opteron 64 bit hardware we choose UMFPACK 4.3 [27] as subdomain direct solver for the Dirichlet problems as well as for the Neumann problems and the coarse problem. We also use the AMD Math Core Library (ACML), which is specifically tuned for the AMD64 architecture. On Intel processors, wherever the Intel Fortran 90 compiler is available, we use MUMPS. On these processors we also use the Intel Math Kernel Library (MKL).

3.5 Numerical Results

We apply our implementation of FETI-DP using a change of basis to different problems. First, we consider the standard benchmark problem of an

elastic cube divided into smaller regular cubes as subdomains. As a second example we consider three different mechanical parts from an industrial application, and finally, we apply our algorithm to a cancellous bone geometry, see Sections 3.5.1, 3.5.2, and 3.5.3. The domains considered in the second and third subsections are decomposed in irregularly shaped substructures by using ParMetis [70]. In all problems, we use linear tetrahedral finite elements

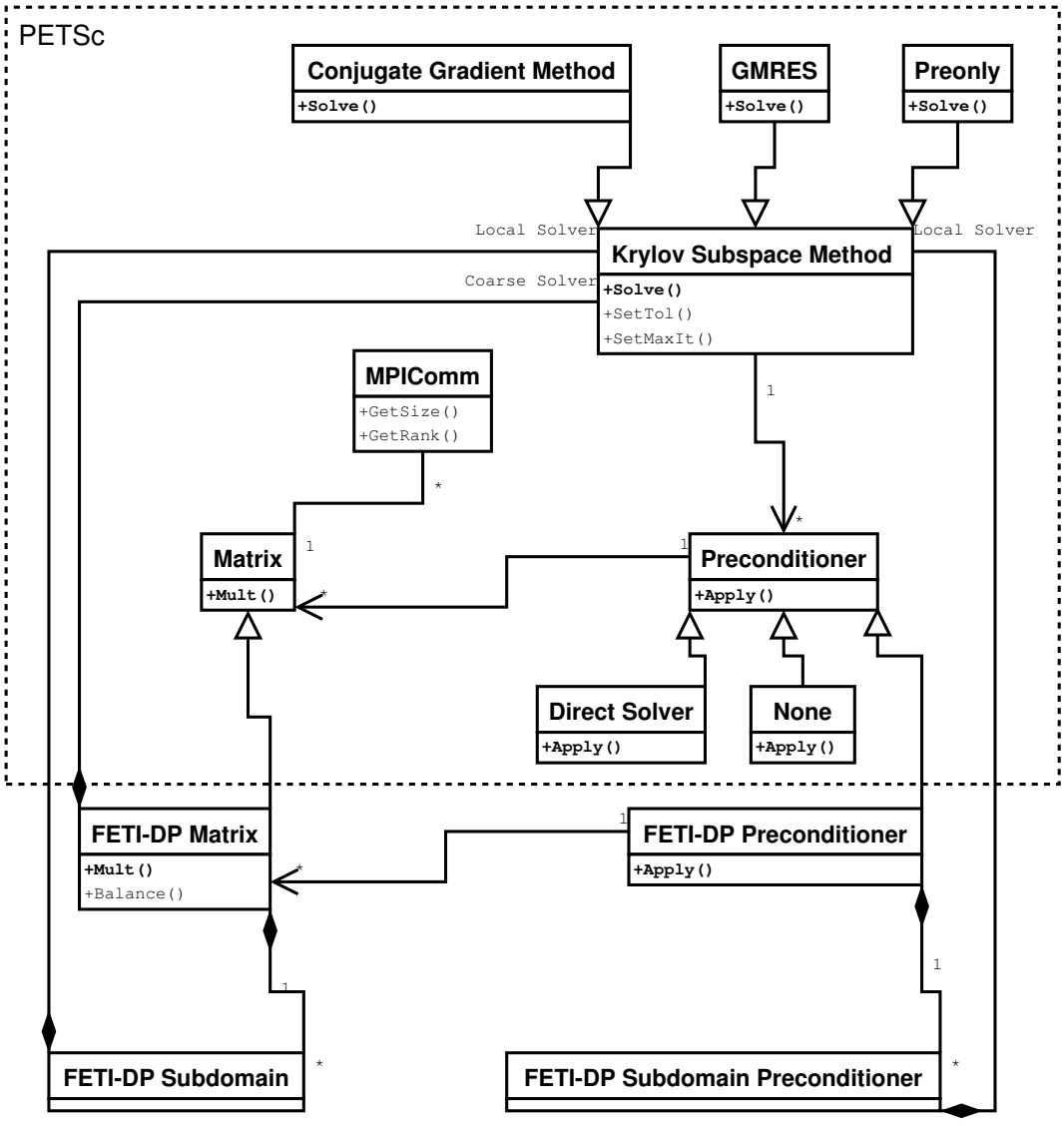

Figure 3.15: *The design of the solver.*

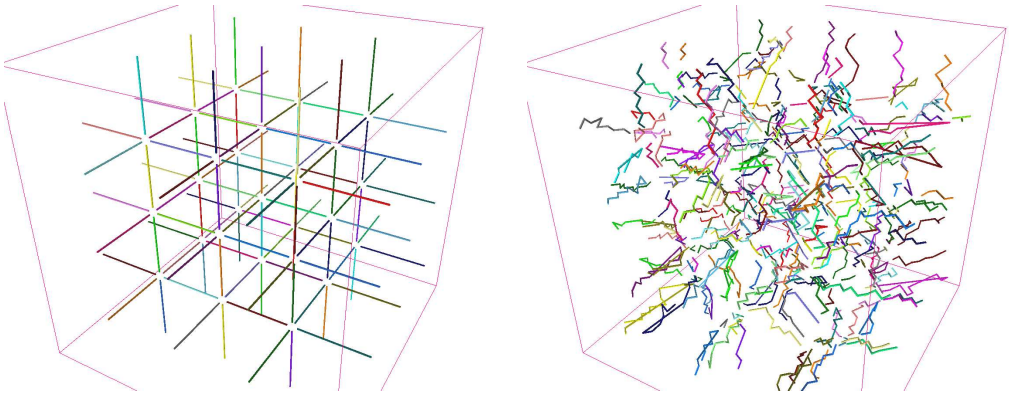

Figure 3.16
Edges as used in the implementation (Definition 3.3.1). Decomposition of the unit cube (446 631 d.o.f.). Structured (left) versus irregular (right, domain decomposition by ParMetis).

and, for simplicity, a Young's modulus of $E = 210$ and a Poisson's ratio of $\nu = 0.29$ throughout. In all of our experiments we use Algorithm D_E, i.e. our choice of primal constraints is given by edge averages on all edges, without any primal vertices. Here, we always constrain all three averages on a primal edge. In all of our computations, we make all edges primal that we obtain from Definition 3.3.1; additionally, in Sections 3.5.2 and 3.5.3, we add some edges obtained by Definition 3.3.1 using N_x^C.

All computations of Sections 3.5.1 and 3.5.2 were carried out on *Jazz*, a 350 node computing cluster operated by the Mathematics and Computer Science Division at Argonne National Laboratory, USA. The cluster consists of 2.4 GHz Intel Xeon processors with 1 or 2 GB of memory each and uses a Myrinet connection. The numerical results given in Table 3.17, Figure 3.22 and in Section 3.5.3 have been carried out on a 16 processor Linux computing cluster in Essen with eight dual 2.2 GHz AMD Opteron nodes and 4 GB of memory for each processor. This cluster uses a Gigabit Ethernet connection.

3.5.1 Model Problem

In this section, as a benchmark model problem, we consider a homogeneous, isotropic, linearly elastic cube, which is clamped at one side, while all other parts of the boundary have homogeneous natural boundary conditions. A volume force is applied, which defines the right hand side.

In order to analyze the numerical and parallel scalability of our FETI-DP

algorithm we report on two different series of experiments. In our first set of runs we keep the dimension of the local problems, and H/h, fixed and increase the number N of subdomains and thus the overall problem size; see Tables 3.5 and 3.6. In a second series of experiments we keep a fixed number N of subdomains and increase the size of the local problems, and H/h, resulting in a smaller h and thus a larger overall problem size. The results of these experiments are given in Table 3.7 and Figure 3.17. In all sets of experiments we use as a stopping criterion the relative reduction of the preconditioned dual residual by 10^{-7}.

In the numerical results obtained for a fixed subdomain size, reported in Tables 3.5 and 3.6, the cube is partitioned into smaller cubes with $H/h = 14$, which results in 8 232 degrees of freedom for each subdomain. In Tables 3.5 and 3.6 we denote the degrees of freedom of the original, assembled problem by "D.o.f." and those of the coarse problem by "Coarse". We first present results using the sparse direct solver built into PETSc [9, 8, 10] for solving the coarse problem and the local problems; see Table 3.5.

Proc.	N	$1/h$	**D.o.f.**	**Coarse**	$\lambda_{\min}$	$\lambda_{\max}$	**It.**	**Time**
1	8	27	59 049	18	1.03	12.94	18	137s
8	64	53	446 631	324	1.03	10.45	23	171s
27	216	79	1 479 117	1 350	1.04	10.32	23	188s
64	512	105	3 472 875	3 528	1.04	10.31	23	192s
125	1 000	131	6 744 273	7 290	1.04	10.30	23	226s

Table 3.5
Results for a cube divided into smaller cubes with a fixed subdomain size of 8 232 d.o.f. using PETSc as the subdomain and coarse sparse direct solver.

Next, we present a set of experiments using the sparse direct solver package MUMPS 4.3.2 [5] together with the optimized BLAS libraries for the Xeon architecture to solve the coarse problem and the local problems; see Table 3.6. Both set of experiments show that our FETI-DP algorithm using only edge averages as the primal constraints yields a numerical and parallel scalable domain decomposition method. As a further result of this comparison, we see that using MUMPS as a direct solver accelerates our method by almost a factor of three in terms of CPU time.

The numerical results for a fixed number of subdomains with increasing size are obtained for the unit cube divided into $N = 4 \times 4 \times 4 = 64$ subdo-

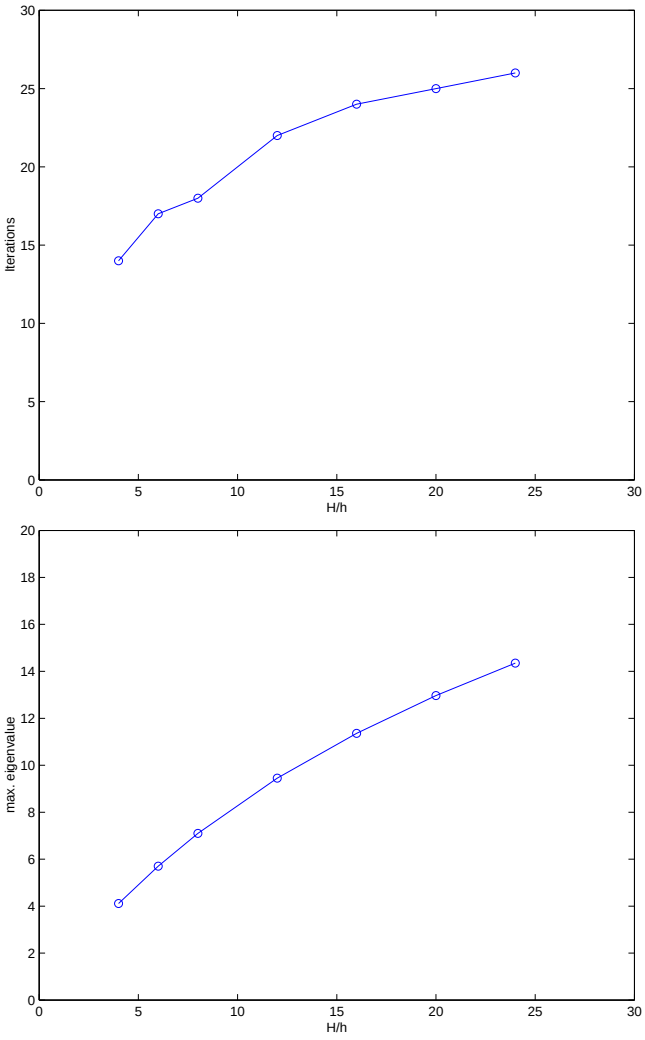

Figure 3.17
Growth of number of iterations (left) and λ_{max} (right) of Algorithm D_E for the unit cube with variable H/h and $4 \times 4 \times 4$ subdomains.

Proc.	N	$1/h$	D.o.f.	Coarse	$\lambda_{\min}$	$\lambda_{\max}$	It.	Time
4	64	53	446 631	324	1.03	10.45	23	60.3s
32	512	105	3 472 875	3 528	1.04	10.31	23	69.2s
108	1 728	157	11 609 679	13 068	1.04	10.30	23	78.8s

Table 3.6
Results for a cube divided into smaller cubes with a fixed subdomain size of 8 232 *d.o.f. using MUMPS as the subdomain and coarse sparse direct solver.*

D.o.f.	H/h	It.	$\lambda_{\min}$	$\lambda_{\max}$
6 591	4	14	1.03	**4.11**
27 783	6	17	1.03	**5.70**
73 167	8	18	1.03	**7.10**
273 375	12	22	1.03	**9.45**
680 943	16	24	1.03	**11.36**
1 369 599	20	25	1.04	**12.97**
2 413 071	24	26	1.04	**14.35**

Table 3.7
Results for $4 \times 4 \times 4$ *subdomains of increasing size. The iteration count is given for a relative reduction of the preconditioned dual residual by* 10^{-7}.

mains. We keep $H = 1/4$ fixed and vary H/h between 4 and 24. Algorithm D_E has a coarse problem size of $\dim(\widetilde{S}_{\Pi\Pi}) = 324$. The growth of the number of iterations and of the largest eigenvalue of Algorithms D_E is shown in Figure 3.17.

All of the numerical results in this subsection confirm the theoretical condition number estimate given in Theorem 3.3.1 and show the good performance of Algorithm D_E.

3.5.2 Industrial Applications

In this section, we apply our FETI-DP algorithm to three different industrial finite element problems, denoted by mechanical parts A, B, and C. Here, we have to use irregular decompositions using a graph partitioner. We therefore have irregular edges, see Figure 3.16 (p. 80).

The direct subdomain factorizations as well as the coarse problem fac-

Proc.	2	4	8	16
FETI-DP D_E/MUMPS	62s	33s	**20s**	**11s**
MUMPS	**47s**	**27s**	26s	19s

Table 3.8
Parallel scalability results for mechanical part A: CPU times for FETI-DP using MUMPS as the local and coarse sparse direct solver and for MUMPS applied to the undecomposed problem.

torization are performed, again, by using the sparse direct solver MUMPS and the optimized BLAS libraries for the Xeon architecture. To compare the parallel performance of our dual-primal FETI domain decomposition method Algorithm D_E to that of the parallel sparse direct solver provided by MUMPS, we also provide CPU timings for MUMPS applied to the assembled and undecomposed problem on the same machine.

The first problem, mechanical part A, cf. Figure 3.18, has been discretized by 208 536 linear tetrahedral finite elements yielding a global number of 187 539 d.o.f. In the reported experiments the mechanical part A is partitioned into $N = 16$ subdomains using ParMetis. The FETI-DP algorithm using only edge averages as primal constraints needed 29 iterations for a relative reduction of the preconditioned dual residual by 10^{-7}. The size of the coarse problem is 393 d.o.f. The smallest and largest eigenvalues are $\lambda_{\min} = 1.02$ and $\lambda_{\max} = 19.63$, respectively. The parallel scalability results on 2 to 16 processors are given in Table 3.8.

The second problem in this subsection, mechanical part B, cf. Figure 3.19, is discretized by 581 394 linear tetrahedral finite elements resulting in a global number of 380 709 d.o.f. It is partitioned into $N = 64$ subdomains using ParMetis. In all of the experiments reported in Table 3.9 our FETI-DP algorithm using only edge averages as primal constraints needed 29 iterations for a relative reduction of the preconditioned dual residual by 10^{-7}. The size of the coarse problem is 1 020 d.o.f. The smallest and largest eigenvalues are $\lambda_{\min} = 1.03$ and $\lambda_{\max} = 33.85$, respectively. The parallel scalability results on 4 to 64 processors are given in Table 3.9.

The third problem in this subsection, mechanical part C, cf. Figure 3.20, is discretized by 1 291 933 linear tetrahedral finite elements yielding a global number of 841 836 d.o.f. It is partitioned into $N = 64$ subdomains using ParMetis. In all of the experiments reported in Table 3.10 the number of iterations is 32 for a relative reduction of the preconditioned dual residual by

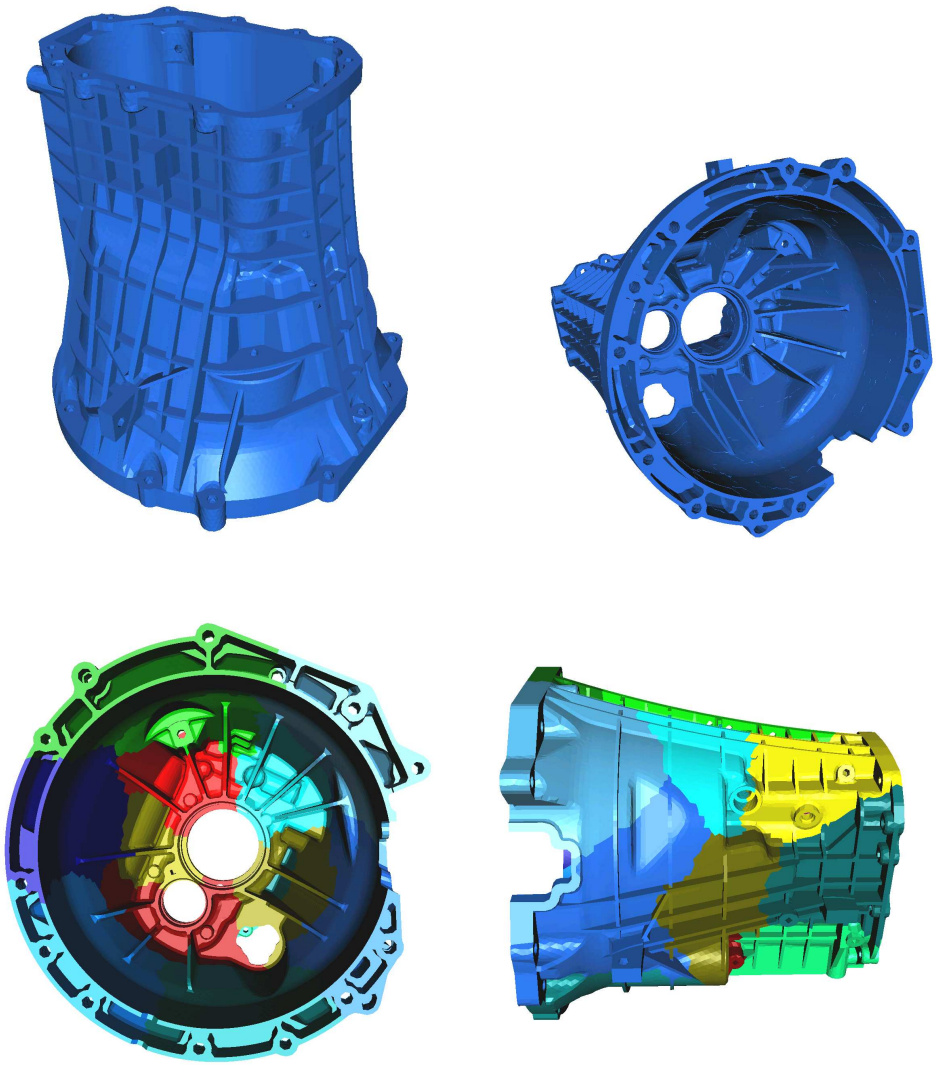

Figure 3.18
Mechanical part A; courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany.

Proc.	4	8	16	32	64
FETI-DP D_E/MUMPS	**60s**	**33s**	**18s**	**11s**	**6s**
MUMPS	113s	156s	103s	86s	90s

Table 3.9
Parallel scalability results for mechanical part B: CPU times for FETI-DP using MUMPS as the local and coarse sparse direct solver and for MUMPS applied to the undecomposed problem.

10^{-7}. The size of the coarse problem is 957 d.o.f. The smallest and largest eigenvalues are $\lambda_{\min} = 1.03$ and $\lambda_{\max} = 27.77$, respectively. The parallel scalability results on 16 to 64 processors are given in Table 3.10.

Proc.	16	32	64
FETI-DP D_E/MUMPS	**49s**	**26s**	**17s**
MUMPS	failed	1177s	889s

Table 3.10
Parallel scalability results for mechanical part C: CPU times for FETI-DP with 64 subdomains using MUMPS as the local and coarse sparse direct solver and for MUMPS applied to the undecomposed problem.

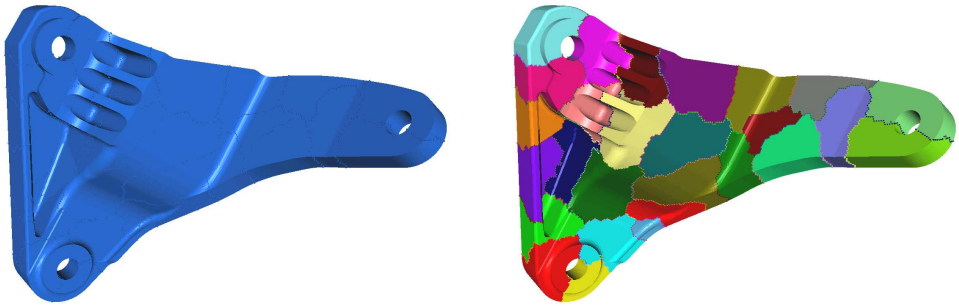

Figure 3.19
Mechanical part B; courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany.

In Table 3.11 we also report results for mechanical part C decomposed into 512 subdomains. In these experiments, the largest eigenvalue is $\lambda_{\max} = 21.36$ and the smallest eigenvalue $\lambda_{\min} = 1.03$. The coarse problem size was chosen as 7 500 d.o.f. The number of iterations is 32 for a relative reduction of the preconditioned dual residual of 10^{-7}.

Proc.	**8**	**16**	**32**	**64**
Time	58s	29s	15.4s	9.3s

Table 3.11
Parallel scalability results for mechanical part C: CPU times for FETI-DP with 512 subdomains using MUMPS as the local and coarse sparse direct solver.

From our numerical experiments we see that the FETI-DP algorithm using only edge averages as primal constraints yields a parallel scalable domain decomposition method also for problems from industrial applications using irregularly shaped substructures. The FETI-DP algorithm is, in terms of CPU time, always faster than the sparse direct solver applied to the undecomposed problem, except for the smallest problem, mechanical part A, and there only for 2 and 4 processors.

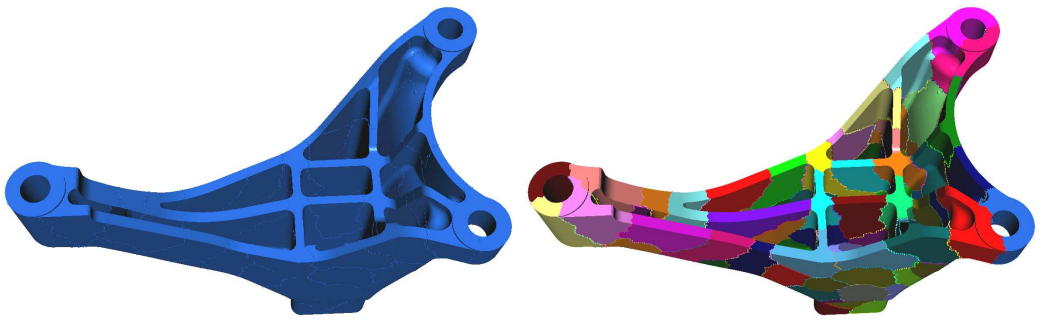

Figure 3.20
Mechanical part C; courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany.

3.5.3 Cancellous Bone

In this subsection, we present results of our FETI-DP algorithm applied to a linearly elastic domain from a cancellous bone geometry. We show that our method is robust also for complicated geometries with many holes and thin structures as they usually appear in cancellous/trabecular bone. For simplicity, we only consider isotropic linear elasticity. For a full study of cancellous bone, we should use linearly elastic orthotropic or even nonlinear material models; this will be considered in forthcoming work. The bone geometry is discretized by 907 609 linear tetrahedral finite elements resulting in a global number of 620 730 d.o.f. The mesh is obtained using a marching cubes algorithm to generate a surface mesh and Netgen [116] to produce a volume mesh from it. It is partitioned into $N = 96$ subdomains using ParMetis. In all of our experiments reported in Table 3.12, the number of iterations is 51 for a relative reduction of the preconditioned dual residual by 10^{-10}. The size of the coarse problem is 1 602 d.o.f. The smallest and largest eigenvalues are $\lambda_{\min} = 1.02$ and $\lambda_{\max} = 43.57$, respectively. The parallel scalability results on 1 to 16 processors are given in Table 3.12. As sparse direct solvers for the coarse and the local problems, we use UMFPACK 4.3 [27].

In Figure 3.22 we show the convergence history for the cancellous bone geometry as well as for mechanical part A, mechanical part B, and for the cube benchmark problem with $H/h = 14$ and $h = 1/105$. We compute the true relative residual $\|F\lambda^n - d\|_2/\|F\lambda^0 - d\|_2$ explicitly in each step, see Figure 3.22. Let us note that the true residual is only used in the experiments for the results shown in Figure 3.22. Here, we use the true residual for the computation of the results to avoid possible inaccuracies in the recursive computation used in the conjugate gradient algorithm when the residual is approaching machine precision. In all cases the algorithm is able to reduce the true residual by fourteen orders of magnitude.

Proc.	**1**	**2**	**4**	**8**	**16**
Time	524s	312s	170s	90s	46s

Table 3.12: *Parallel scalability of FETI-DP for cancellous bone geometry.*

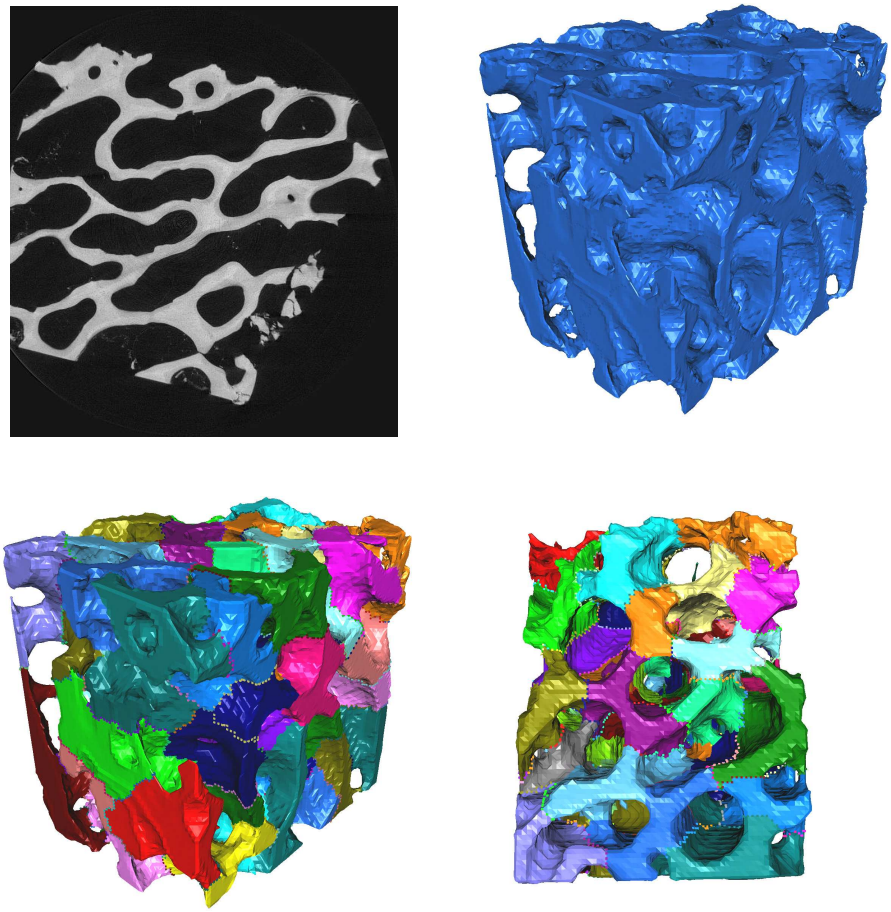

Figure 3.21

Cancellous bone. Upper left: Cross section of X-ray Computer Tomography. Upper right: Finite element discretization with 907 609 *tetrahedra and* 620 730 *d.o.f. Lower left and right: Different views of a decomposition of the bone into subdomains by ParMetis.*

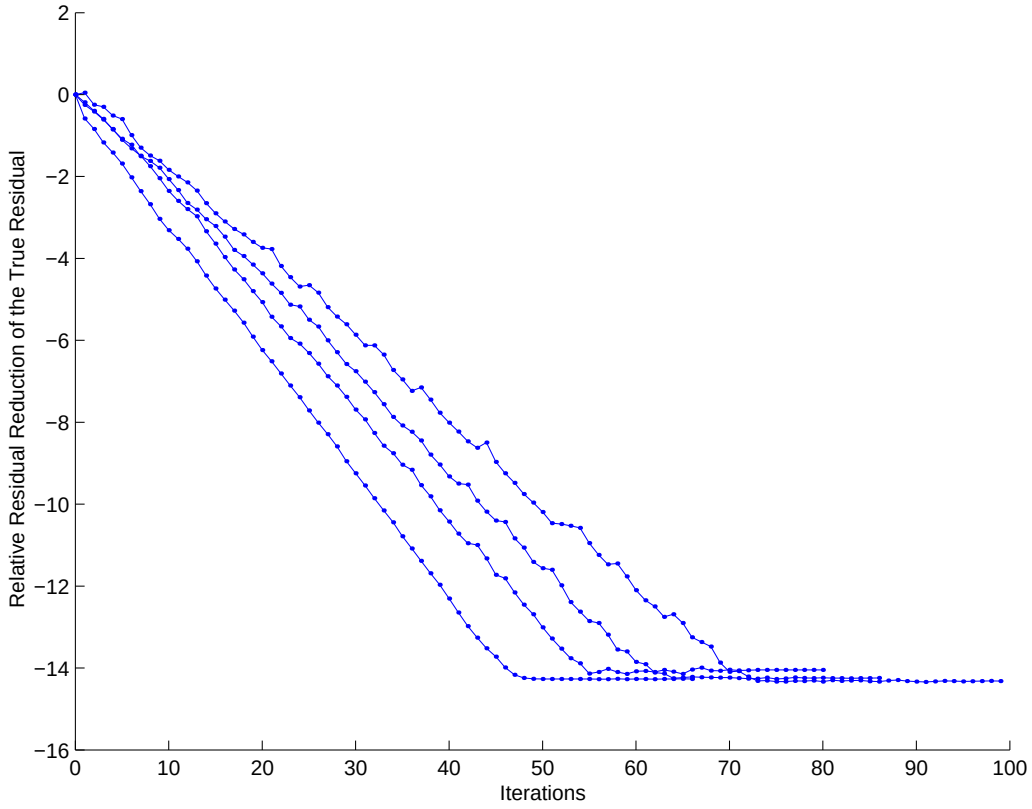

Figure 3.22

History of relative residual reductions of the true residual. From left to right: Cube benchmark problem (105^3), mechanical part A, mechanical part B, cancellous bone geometry.

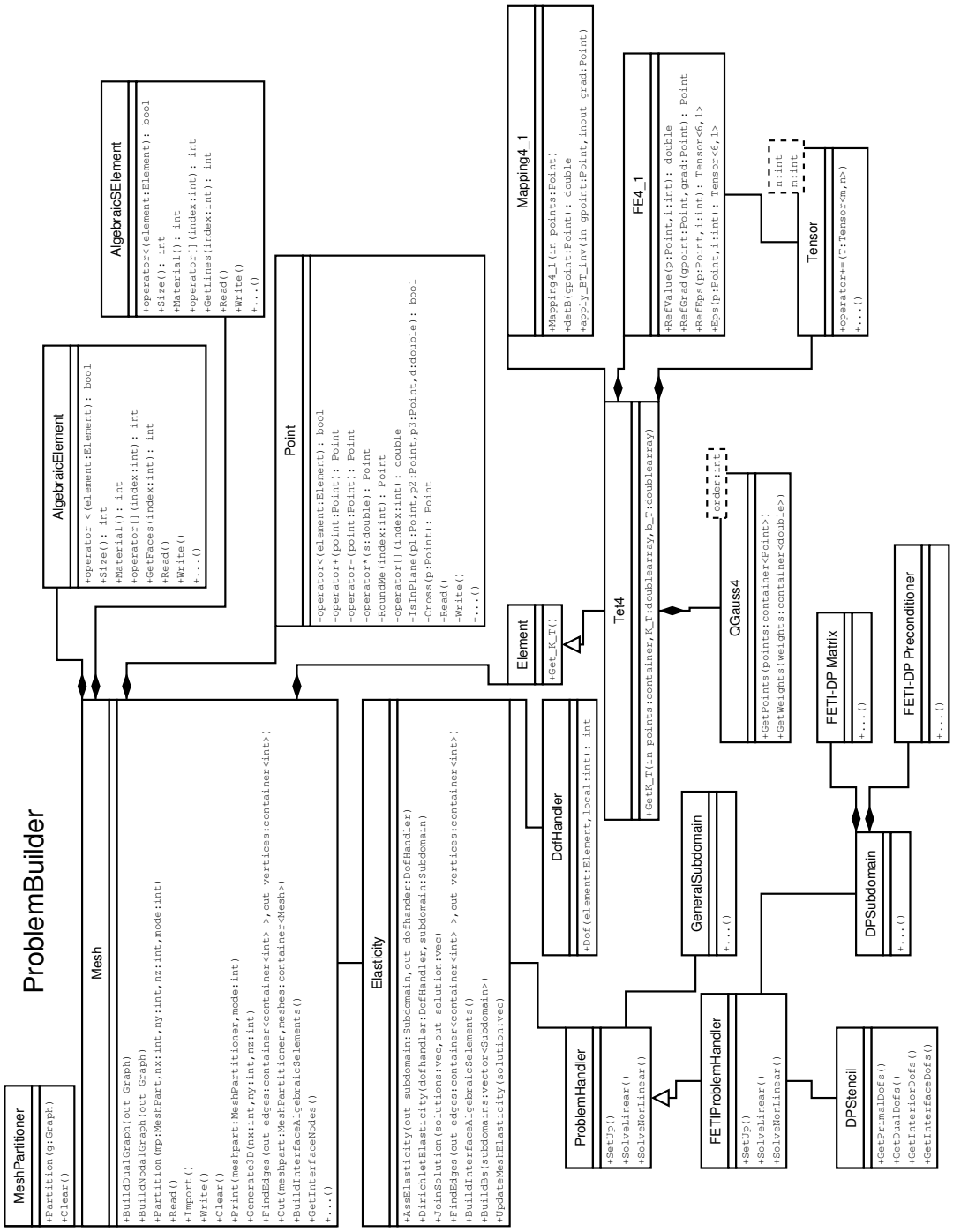

Figure 3.23

The design of the problem module (partial view, function parameters given where considered helpful).

Chapter 4

Inexact FETI-DP Methods

FETI-DP methods obtain their numerical and parallel scalability from the use of a coarse problem which is very small compared to the overall problem: The algorithms defined in Chapter 3 lead to a coarse problem of the order of one tenth of a percent of the total problem size. The coarse problem is traditionally solved exactly by the use of a direct solver. Nevertheless, if a very large number of subdomains is used or if the problem requires the use of a very large coarse space, solving the coarse problem directly may not be feasible anymore.

Here, we present a family of methods, based on different versions of the FETI-DP saddle point formulation, which allow for the use of inexact solvers for the FETI-DP coarse problem. In one variant, which iterates on the complete saddle point system, it is also possible to solve the local Neumann problems inexactly; see also Klawonn and Widlund [81] for a related approach for the classical, onelevel FETI method. We note that in the approach in [81], the coarse problem has to be solved exactly. The resulting preconditioned systems in this work are based on appropriate block triangular preconditioners and can either be solved by GMRES or by any other Krylov space method suitable for nonsymmetric linear systems, e.g. BiCGSTAB or QMR. We also discuss a positive definite reformulation which can be solved by the method of conjugate gradients. This approach dates back to work on preconditioners for saddle point problems by Bramble and Pasciak [17]; see also Klawonn [73] and Dohrmann and Lehoucq [32]. The inexact solution of the local Dirichlet problems is always possible with FETI methods, e.g. by the use of the nonoptimal, lumped preconditioner. Let us note that the algorithms presented here also allow for optimal, inexact Dirichlet subdomain solvers. The presentation in this chapter is based on Klawonn and Rheinbach [75].

An approach to solve the BDDC coarse problem inexactly has successfully been suggested and analyzed by Tu [138, 139]. Such an approach is more

straightforward for BDDC methods, since the coarse problem is built into the preconditioner and not into the system matrix. For very recent work on approximate subdomain solvers for BDDC methods, see Li and Widlund [91]. In Gosselet [56] an interesting hybrid domain decomposition method is considered and applied to multifield problems which iterates on a linear system consisting of primal and dual variables at the same time.

In this chapter, we again consider the system of linear elasticity as a model problem. We note that other elliptic partial differential equations could be treated as well using the methods provided here.

4.1 FETI-DP Saddle Point Formulation

Let us consider the FETI-DP master system

$$\begin{bmatrix} \widetilde{K} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ 0 \end{bmatrix}, \qquad u \in \mathbf{R}^n,\ \lambda \in \mathbf{R}^m, \tag{4.1}$$

where

$$\widetilde{K} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix} = \begin{bmatrix} K_{BB}^{(1)} & & & \widetilde{K}_{B\Pi}^{(1)} \\ & \ddots & & \vdots \\ & & K_{BB}^{(N)} & \widetilde{K}_{B\Pi}^{(N)} \\ \tilde{K}_{\Pi B}^{(1)} & \cdots & \tilde{K}_{\Pi B}^{(N)} & \tilde{K}_{\Pi\Pi} \end{bmatrix},$$

and

$$\tilde{f} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \end{bmatrix} = \begin{bmatrix} f_B^{(1)} \\ \vdots \\ f_B^{(N)} \\ \tilde{f}_\Pi \end{bmatrix},$$

see Chapter 2, from which the solution of the original finite element problem (2.1) can be obtained by joining the solution u in the interface variables $u_\Delta^{(i)}$.

We introduce the notation

$$\mathcal{A}x = \mathcal{F},$$

where

$$\mathcal{A} := \begin{bmatrix} \widetilde{K} & B^T \\ B & 0 \end{bmatrix},\ x := \begin{bmatrix} u \\ \lambda \end{bmatrix},\ \mathcal{F} := \begin{bmatrix} \tilde{f} \\ 0 \end{bmatrix}.$$

Let us write (4.1) in the form

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B_B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}. \tag{4.2}$$

Eliminating u_B by one step of block Gaussian elimination, we obtain the reduced system

$$\begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B}K_{BB}^{-1}B_B^T \\ -B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T & -B_BK_{BB}^{-1}B_B^T \end{bmatrix} \begin{bmatrix} \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{f}_\Pi - \widetilde{K}_{\Pi B}K_{BB}^{-1}f_B \\ -B_BK_{BB}^{-1}f_B \end{bmatrix}, \quad (4.3)$$

where $\widetilde{S}_{\Pi\Pi} = \widetilde{K}_{\Pi\Pi} - \widetilde{K}_{\Pi B}K_{BB}^{-1}\widetilde{K}_{\Pi B}^T$. Here, we will also use the notation

$$\mathcal{A}_r x_r = \mathcal{F}_r,$$

where

$$\mathcal{A}_r = \begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B}K_{BB}^{-1}B_B^T \\ -B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T & -B_BK_{BB}^{-1}B_B^T \end{bmatrix}, \; x_r := \begin{bmatrix} \tilde{u}_\Pi \\ \lambda \end{bmatrix},$$

and

$$\mathcal{F}_r := \begin{bmatrix} \tilde{f}_\Pi - \widetilde{K}_{\Pi B}K_{BB}^{-1}f_B \\ -B_BK_{BB}^{-1}f_B \end{bmatrix}.$$

By also eliminating the primal variables $\tilde{u}_\Pi$, we obtain the reduced system

$$F\lambda = d, \quad (4.4)$$

where

$$\begin{aligned} F &:= B_BK_{BB}^{-1}B_B^T + B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T\widetilde{S}_{\Pi\Pi}^{-1}\widetilde{K}_{\Pi B}K_{BB}^{-1}B_B^T = B\widetilde{K}^{-1}B^T, \\ d &:= B_BK_{BB}^{-1}f_B + B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T\widetilde{S}_{\Pi\Pi}^{-1}(\tilde{f}_\Pi - \widetilde{K}_{\Pi B}K_{BB}^{-1}f_B) = B\widetilde{K}^{-1}\tilde{f}. \end{aligned}$$

The linear system (4.4) is the standard, exact FETI-DP system, see Chapter 2, which is solved using Preconditioned Conjugate Gradients and an appropriate preconditioner M^{-1}.

As usual, we will use a transformation of basis to enforce edge averages as primal constraints, in 2D as well as in 3D. In our experiments, in two dimensions, we will use continuity at all vertices and continuity of all edge averages in order to create a larger coarse problem. This version will be denoted Algorithm B, see [86] where an analogous notation was introduced for three dimensions. In three dimensions, we will use Algorithm D_E, see Chapter 3.

4.2 Exact and Inexact FETI-DP Methods

In the standard, exact FETI-DP methods two different preconditioners are commonly used, the theoretically almost optimal Dirichlet preconditioner M_{FETI_D} and the lumped preconditioner M_{FETI_L}, see (2.4) and (2.5).

The original or standard, exact FETI-DP method, see Chapter 2, is the method of conjugate gradients applied to the symmetric positive definite system

$$F\lambda = d$$

with the preconditioners $M^{-1}_{\mathrm{FETI}_D}$ or $M^{-1}_{\mathrm{FETI}_L}$. We note that only for the Dirichlet preconditioner do we have the polylogarithmic condition number bounds mentioned before; see also [100, 86, 84]. Here, the term "exact" refers to the exact solution of the coarse problem given by $\widetilde{S}_{\Pi\Pi}$ and the exact solution of the local Neumann subdomain problems $K^{(i)}_{BB}$. When the (exact) Dirichlet preconditioner is used, we also have to solve the local Dirichlet problems $K^{(i)}_{II}$ exactly.

Now, we are going to present new, inexact FETI-DP methods by solving the saddle point problems (4.1), (4.2) and (4.3) iteratively, using block triangular preconditioners and a suitable Krylov space method.

For the saddle point problems (4.1), (4.2) and (4.3), we introduce the block triangular preconditioners $\widehat{\mathcal{B}}_L$ and $\widehat{\mathcal{B}}_{r,L}$, respectively, as

$$\widehat{\mathcal{B}}_L^{-1} = \begin{bmatrix} \widehat{K}^{-1} & 0 \\ M^{-1}B\widehat{K}^{-1} & -M^{-1} \end{bmatrix}, \quad \widehat{\mathcal{B}}_{r,L}^{-1} = \begin{bmatrix} \widehat{S}^{-1}_{\Pi\Pi} & 0 \\ -M^{-1}B_B K^{-1}_{BB}\widetilde{K}^T_{\Pi B}\widehat{S}^{-1}_{\Pi\Pi} & -M^{-1} \end{bmatrix},$$

where $\widehat{K}^{-1}$ and $\widehat{S}^{-1}_{\Pi\Pi}$ are assumed to be spectrally equivalent preconditioners for $\widetilde{K}$ and $\widetilde{S}_{\Pi\Pi}$, respectively, with bounds independent of the discretization parameters h, H. The matrix block M^{-1} is assumed to be a good preconditioner for the FETI-DP system matrix F and can be chosen as the Dirichlet or the lumped preconditioners M_D^{-1} and M_L^{-1}, respectively. We will denote the corresponding right preconditioners by the subscript R, i.e. we have $\widehat{\mathcal{B}}_R = \widehat{\mathcal{B}}_L^T$ and $\widehat{\mathcal{B}}_{r,R} = \widehat{\mathcal{B}}_{r,L}^T$.

We note that $\widehat{K}^{-1}$ can also be defined using the following exact factorization of $\widetilde{K}^{-1}$, i.e.

$$\begin{bmatrix} K_{BB} & \widetilde{K}^T_{\Pi B} \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix}^{-1} = \begin{bmatrix} I & -K^{-1}_{BB}\widetilde{K}^T_{\Pi B} \\ 0 & I \end{bmatrix} \begin{bmatrix} K^{-1}_{BB} & 0 \\ 0 & \widetilde{S}^{-1}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\widetilde{K}_{\Pi B}K^{-1}_{BB} & I \end{bmatrix}. \tag{4.5}$$

In this case, $\widetilde{K}_{\Pi B}K^{-1}_{BB} =: \overline{K}_{\Pi B}$ is built explicitly in a preprocessing step since we need it to form $\widetilde{S}_{\Pi\Pi}$. To obtain a preconditioner $\widehat{K}^{-1}$, we can now replace K^{-1}_{BB} and $\widetilde{S}^{-1}_{\Pi\Pi}$ by good preconditioners $\widehat{K}^{-1}_{BB}$ and $\widehat{S}^{-1}_{\Pi\Pi}$. This yields the preconditioner

$$\widehat{K}^{-1} = \begin{bmatrix} I & -\overline{K}^T_{\Pi B} \\ 0 & I \end{bmatrix} \begin{bmatrix} \widehat{K}^{-1}_{BB} & 0 \\ 0 & \widehat{S}^{-1}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\overline{K}_{\Pi B} & I \end{bmatrix}. \tag{4.6}$$

We note that the application of $\widehat{K}^{-1}$ to a vector only involves one application of $\widehat{K}_{BB}^{-1}$ and $\widehat{S}_{\Pi\Pi}^{-1}$ each. Such a factorization was also the basis for iterative substructuring methods with inexact Dirichlet solvers, see, e.g., Smith, Bjørstad, and Gropp [126, Chapter 4.4] or Toselli and Widlund [136, Chapter 4.3] and the references given therein.

It is also possible to use exact local solvers, i.e. $\widehat{K}_{BB}^{-1} = K_{BB}^{-1}$, and to solve only the coarse problem inexactly. This variant is closely related to preconditioning the reduced system (4.3) by an appropriate block triangular preconditioner.

Instead of using the factorization (4.6), a preconditioner can also be applied directly to $\widetilde{K}$, see Section 4.8.4. This results in an algorithm where the subdomain solves as well as the coarse grid solve are inexact.

Our inexact FETI-DP methods are now given by using a Krylov space method for nonsymmetric systems, e.g. GMRES, to solve the preconditioned systems

$$\widehat{\mathcal{B}}_L^{-1}\mathcal{A}x = \widehat{\mathcal{B}}_L^{-1}\mathcal{F}$$

and

$$\widehat{\mathcal{B}}_{r,L}^{-1}\mathcal{A}_r x_r = \widehat{\mathcal{B}}_L^{-1}\mathcal{F}_r,$$

respectively.

Let us note that we can also use a positive definite reformulation of the two preconditioned systems which allows for the use of conjugate gradients. For this reformulation, a special inner product and a scaling of the preconditioners $\widehat{K}$ and $\widehat{S}_{\Pi\Pi}$ have to be used, see Sections 4.3 and 4.4 for further details.

4.3 Block Triangular Preconditioners for Symmetric Saddle Point Problems

In this section, we review some theoretical results for block triangular preconditioners applied to symmetric saddle point problems. This theory will then be used in the next section to derive convergence estimates for the full and the reduced preconditioned system. In general, using block triangular preconditioners leads to nonsymmetric preconditioned systems, even when the original saddle point problem is symmetric. Thus, Krylov space methods which are well suited for nonsymmetric linear systems have to be chosen, e.g. GMRES, BiCGSTAB, QMR or variants of these methods. In some cases, the preconditioned system is symmetric positive definite in a certain inner product. Then, a conjugate gradient method can be used, see Bramble and

Pasciak [17]. Let us note that the theory for deriving a priori GMRES convergence bounds for block triangular preconditioners is not complete and still an area of research. In contrast to conjugate gradient methods, eigenvalue bounds are in general not sufficient for convergence estimates of GMRES. Our presentation in this section is based on Klawonn [73].

We consider a mixed linear system of the form

$$\mathcal{A}x = \mathcal{F}, \tag{4.7}$$

where we have

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}, \; x = \begin{bmatrix} u \\ \lambda \end{bmatrix}, \; \mathcal{F} = \begin{bmatrix} f \\ g \end{bmatrix}.$$

We assume that $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, $C \in \mathbb{R}^{m\times m}$ a symmetric positive semidefinite matrix, and $B \in \mathbb{R}^{m\times n}$ a matrix with full rank. Furthermore, we define left and right block triangular preconditioners

$$\widehat{\mathcal{B}}_L = \begin{bmatrix} \hat{A} & 0 \\ B & -\widehat{C} \end{bmatrix}, \quad \widehat{\mathcal{B}}_R = \begin{bmatrix} \hat{A} & B^T \\ 0 & -\widehat{C} \end{bmatrix}.$$

Here, we assume that there exist constants $\alpha_0, \alpha_1 > 0$ such that

$$\alpha_0 \; u^T \hat{A} u \leq u^T A u \leq \alpha_1 \; u^T \hat{A} u \quad \forall u \in \mathbb{R}^n \tag{4.8}$$

and constants $\gamma_0, \gamma_1 > 0$ such that

$$\gamma_0 \; \lambda^T \widehat{C} \lambda \leq \lambda^T S_C \lambda \leq \gamma_1 \; \lambda^T \widehat{C} \lambda \quad \forall \lambda \in \mathbb{R}^m, \tag{4.9}$$

where the Schur complement S_C is defined as $S_C := C + BA^{-1}B^T$.

In our analysis, we only consider the case of inexact preconditioners $\hat{A}$, i.e. we exclude the case $\hat{A} = A$. In our application on FETI-DP methods, the exact case relates to the standard, exact FETI-DP method, and therefore, we do not have to analyze it here. Nevertheless, an exact solver for A can be applied with GMRES; see Klawonn [73] and Simoncini [125] for numerical results and eigenvalue bounds.

To the best of our knowledge, the first GMRES convergence analysis for block triangular preconditioners applied to symmetric saddle point problems was given in Klawonn [72, 73], where the following assumption was made for the preconditioner $\hat{A}$,

$$1 < \alpha_0 \leq a_1, \tag{4.10}$$

which can be always obtained by an appropriate scaling. We will briefly review those results using our notation. We first introduce the symmetric

positive definite matrices $\mathcal{H}$ and $\widetilde{\mathcal{H}}$,

$$\mathcal{H} = \begin{bmatrix} A - \hat{A} & 0 \\ 0 & \widehat{C} \end{bmatrix}, \quad \widetilde{\mathcal{H}} = \begin{bmatrix} A & 0 \\ 0 & S_C \end{bmatrix}.$$

From a direct calculation, we obtain the symmetric matrix

$$\mathcal{H}\widehat{\mathcal{B}}_L^{-1}\mathcal{A} = \begin{bmatrix} A\hat{A}^{-1}A - A & (A - \hat{A})\hat{A}^{-1}B^T \\ B\hat{A}^{-1}(A - \hat{A}) & C + B\hat{A}^{-1}B^T \end{bmatrix}.$$

To apply the theory proven in [73], we note that the equality $\mathcal{H}\widehat{\mathcal{B}}_L^{-1}\mathcal{A} = \mathcal{A}\widehat{\mathcal{B}}_R^{-1}\mathcal{H}$ holds; see also [73, Remark 2]. The next lemma is proven in [73, Lemma 3.3].

Lemma 4.3.1 *There exist positive constants $\widetilde{C}_0, \widetilde{C}_1$ such that*

$$\widetilde{C}_0 \; x^T\widetilde{\mathcal{H}}x \le x^T\mathcal{H}\widehat{\mathcal{B}}_L^{-1}\mathcal{A}x \le \widetilde{C}_1 \; x^T\widetilde{\mathcal{H}}x \quad \forall x \in \mathbb{R}^{n+m},$$

where $\widetilde{C}_0 = \min\{(\alpha_0 - 1), 1\}/3$ and $\widetilde{C}_1 = 3\max\{(\alpha_1 - 1), 1\}$.

Using (4.8), (4.9), and (4.10), we obviously have the following spectral equivalence:

$$\min\{\frac{\alpha_1}{\alpha_1 - 1}, \gamma_0\} \; x^T\mathcal{H}x \le x^T\widetilde{\mathcal{H}}x \le \max\{\frac{\alpha_0}{\alpha_0 - 1}, \gamma_1\} \; x^T\mathcal{H}x \quad \forall x \in \mathbb{R}^{n+m}. \tag{4.11}$$

Combining Lemma 4.3.1 and (4.11), we obtain Lemma 4.3.2, see also [73, Lemma 3.4].

Lemma 4.3.2 *We have*

$$C_0 \; x^T\mathcal{H}x \le x^T\mathcal{H}\widehat{\mathcal{B}}_L^{-1}\mathcal{A}x \le C_1 \; x^T\mathcal{H}x \quad \forall x \in \mathbb{R}^{n+m}$$

with positive constants $C_0 = \left(\frac{1}{3}\min\{(\alpha_0 - 1), 1\}\min\{\frac{\alpha_1}{\alpha_1 - 1}, \gamma_0\}\right)$ and $C_1 = \left(3\max\{(\alpha_1 - 1), 1\}\max\{\frac{\alpha_0}{\alpha_0 - 1}, \gamma_1\}\right)$.

From this lemma immediately follows that the eigenvalues of $\widehat{\mathcal{B}}_L^{-1}\mathcal{A}$ are real, positive, and contained in the interval $[C_0, C_1]$; cf. also [73, Theorem 3.5]. We can now use the bounds given in Lemma 4.3.2 to provide a convergence bound for GMRES minimizing the residual in an arbitrary norm equivalent to the $\mathcal{H}$-norm; see [73, Theorem 3.7] where this result is given for right preconditioning with $\widehat{\mathcal{B}}_R^{-1}$ and the $\mathcal{H}^{-1}$-inner product. The result is based on

the fact that the method of conjugate residuals and GMRES both minimize the same residual in the norm used, and that for symmetric positive definite matrices, a convergence bound for the method of conjugate residuals can be given in terms of the condition number of the preconditioned system; see [73, Theorem 3.7] for further details.

Theorem 4.3.1 *Let $\widehat{\mathcal{H}}$ be a symmetric positive definite matrix such that $\bar{C}_0\mathcal{H} \leq \widehat{\mathcal{H}} \leq \bar{C}_1\mathcal{H}$ with positive constants $\bar{C}_0, \bar{C}_1$. Then, we have*

$$\frac{\|r^{(k)}\|_{\widehat{\mathcal{H}}}}{\|r^{(0)}\|_{\widehat{\mathcal{H}}}} \leq \frac{\bar{C}_1}{\bar{C}_0} 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k,$$

where $r^{(0)}$ and $r^{(k)}$ are the initial and kth residual of GMRES, respectively, and $\kappa := \kappa(\widehat{\mathcal{B}}_L^{-1}\mathcal{A}) \leq \frac{\bar{C}_1}{\bar{C}_0}$ is the condition number of $\widehat{\mathcal{B}}_L^{-1}\mathcal{A}$ in the $\mathcal{H}$-inner product.

Let us note that for the block triangular preconditioner, to the best of our knowledge, no complete theory exists for a priori GMRES convergence bounds in the Euclidean norm. Nevertheless, the Euclidean inner product is usually used to implement this preconditioning approach with GMRES. Recently, Simoncini [125] has given an eigenvalue analysis of block triangular preconditioners with right preconditioning without the scaling assumption (4.10). The bounds given in [125] also depend on $\alpha_0, \alpha_1, \gamma_0$, and γ_1.

Since $\widehat{\mathcal{B}}_L^{-1}\mathcal{A}$ is symmetric positive definite in the $\mathcal{H}$-inner product, we can also apply the method of conjugate gradients using this special inner product; see Bramble and Pasciak [17] or Dohrmann and Lehoucq [32]. Since this is a nonstandard implementation of the cg-method, we provide a version of this algorithm to solve $\widehat{\mathcal{B}}_L^{-1}\mathcal{A}x = \widehat{\mathcal{B}}_L^{-1}\mathcal{F}$ in Figure 4.1. Here, `xstart` is our initial guess. We note that due to this special implementation, no application of $\widehat{C}$ or $\hat{A}$ is needed. This is important since in our applications, we are usually only able to apply $\widehat{C}^{-1}$ and $\hat{A}^{-1}$ to a vector.

It is well-known that a convergence bound for conjugate gradients can be given in terms of the square root of the spectral condition number of the preconditioned system. From Lemma 4.3.2 we immediately obtain an upper bound for the spectral condition number of $\widehat{\mathcal{B}}_L^{-1}\mathcal{A}$.

4.4 Analysis of the Preconditioners

In this section, we will apply the general theory for block triangular preconditioners presented in Section 4.3 to our inexact FETI-DP methods given in Section 4.2. We only have to identify the matrix blocks in our inexact

$$
\begin{aligned}
\texttt{z} &= \widehat{\mathcal{B}}_L^{-1} \cdot (\mathbf{f} - \mathcal{A} \cdot \texttt{xstart}) \\
\texttt{Hz} &= \mathcal{H}\widehat{\mathcal{B}}_L^{-1} \cdot (\mathbf{f} - \mathcal{A} \cdot \texttt{xstart}) \\
\texttt{p} &= \texttt{z} \\
\texttt{zHz} &= \langle \texttt{z}, \texttt{Hz} \rangle \\
\texttt{lzl} &= \|\texttt{z}\| \\
\text{Until } \|\texttt{z}\|/\texttt{lzl} &< \text{eps} \\
\texttt{Ap} &= \mathcal{A} \cdot \texttt{p} \\
\texttt{HBAp} &= \mathcal{H}\widehat{\mathcal{B}}_L^{-1} \cdot \texttt{Ap} \\
\alpha &= \texttt{zHz} / \langle \texttt{HBAp}, \texttt{p} \rangle \\
\texttt{x} &= \texttt{x} + \alpha \cdot \texttt{p} \\
\texttt{z} &= \texttt{z} - \alpha \widehat{\mathcal{B}}_L^{-1} \cdot \texttt{Ap} \\
\texttt{Hz} &= \texttt{Hz} - \alpha \cdot \texttt{HBAp} \\
\texttt{zHzo} &= \texttt{zHz} \\
\texttt{zHz} &= \langle \texttt{z}, \texttt{Hz} \rangle \\
\beta &= \texttt{zHz}/\texttt{zHzo} \\
\texttt{p} &= \texttt{z} + \beta \cdot \texttt{p}
\end{aligned}
$$

Figure 4.1: *Conjugate gradient algorithm in the $\mathcal{H}$-inner product.*

FETI-DP methods with those in the general presentation and provide concrete estimates for the constants α_0, α_1 in (4.8) and γ_0, γ_1 in (4.9) in order to obtain our convergence estimates.

4.4.1 Preconditioning the Original System (iFETI-DP)

We first consider the original FETI-DP system (4.1). Here, we have

$$\mathcal{A} = \begin{bmatrix} \widetilde{K} & B^T \\ B & 0 \end{bmatrix}, \quad \widehat{\mathcal{B}}_L^{-1} = \begin{bmatrix} \widehat{K}^{-1} & 0 \\ M^{-1}B\widehat{K}^{-1} & -M^{-1} \end{bmatrix}.$$

Hence, we also have

$$A := \widetilde{K}, \;\; \hat{A} := \widehat{K}, \;\; C := 0, \;\; \widehat{C} := M, \;\; S_C := F,$$

and B is the same matrix as in the original FETI-DP method.

We assume that $\widehat{K}$ is a good preconditioner for $\widetilde{K}$ with optimal spectral bounds α_0 and α_1 which are independent of the discretization parameters h and H. Good examples for such preconditioners are based on geometric and algebraic multigrid methods. Let us note that in some of our experiments incomplete Cholesky decompositions are used although the bounds then will not be optimal.

The spectral bounds γ_0 and γ_1 in (4.9) are given by the eigenvalue bounds of the standard, exact FETI-DP method. There exists a constant $C > 0$, independent of h and H such that

$$\lambda^T M \lambda \leq \lambda^T F \lambda \leq C\,(1 + \log(H/h))^2 \lambda^T M \lambda \quad \forall \lambda \in V,$$

where V is the space of Lagrange multipliers, see Chapters 2 and 3.

Thus, we have

$$\gamma_0 := 1, \;\; \gamma_1 := C\,(1 + \log(H/h))^2.$$

From these estimates, we see that, asymptotically, for our inexact FETI-DP method operating on the original system (4.1), we obtain convergence bounds of the same quality as for the standard, exact FETI-DP methods. This holds for GMRES as well as for conjugate gradients.

4.4.2 Preconditioning the Reduced System (irFETI-DP)

We now consider the reduced FETI-DP system (4.3). Therefore, we have

$$\begin{aligned}\mathcal{A}_r &= \begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B}K_{BB}^{-1}B_B^T \\ -B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T & -B_BK_{BB}^{-1}B_B^T \end{bmatrix}, \\ \widehat{\mathcal{B}}_{r,L}^{-1} &= \begin{bmatrix} \widehat{S}_{\Pi\Pi}^{-1} & 0 \\ -M^{-1}B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T\widehat{S}_{\Pi\Pi}^{-1} & -M^{-1} \end{bmatrix},\end{aligned}$$

and we identify $\mathcal{A}$ and $\widehat{\mathcal{B}}_L$ from Section 4.3 with $\mathcal{A}_r$ and $\widehat{\mathcal{B}}_{r,L}$, respectively. Hence, we also have

$$A := \widetilde{S}_{\Pi\Pi},\ \hat{A} := \widehat{S}_{\Pi\Pi},\ C := B_BK_{BB}^{-1}B_B^T,\ \widehat{C} := M,\ B := -B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T\,.$$

As before, we also assume here that $\widehat{S}_{\Pi\Pi}$ is a good preconditioner for $\widetilde{S}_{\Pi\Pi}$ with optimal spectral bounds α_0 and α_1 which are independent of the discretization parameters h, H.

For the Schur complement S_C we have again

$$S_C = C + BA^{-1}B^T = B_BK_{BB}^{-1}B_B^T + B_BK_{BB}^{-1}\widetilde{K}_{\Pi B}^T\widetilde{S}_{\Pi\Pi}^{-1}\widetilde{K}_{\Pi B}K_{BB}^{-1}B_B^T = F.$$

Since $\widehat{C} = M$, we have

$$\gamma_0 := 1,\ \ \gamma_1 := C\,(1+\log(H/h))^2.$$

From these estimates we see that asymptotically, we again obtain convergence bounds of the same quality as for the standard, exact FETI-DP methods and the inexact FETI-DP methods operating on the original system (4.1). As for the latter method, these bounds hold for GMRES as well as for conjugate gradients.

4.5 Performance Considerations

We will use GMRES or CG to solve the systems (4.1) and (2.2) iteratively using the preconditioners $\widehat{\mathcal{B}}_L^{-1}$ and $\widehat{\mathcal{B}}_{r,L}^{-1}$, respectively. Often, in engineering problems, CG with full orthogonalization is used for FETI methods. In these cases using GMRES comes with no extra cost. Note that in this work, in all experiments using conjugate gradients, we always applied standard CG with short recurrence.

We restrict our rough cost estimate to the use of GMRES. The method for the reduced system (2.2) iterates simultaneously on $\tilde{u}_\Pi$ and λ. Since the

dimension of $\tilde{u}_\Pi$ is small, the computational cost spent in the inner products of the Krylov method is comparable to that of the original, exact FETI-DP method. In fact, the dimension of $[\,\tilde{u}_\Pi^T, \lambda^T\,]^T$ is smaller than or equal to the number of Lagrange multipliers in the original (one-level) FETI method [45, 46]; see, e.g., Klawonn and Widlund [83] or Toselli and Widlund [136]. This is a first indication that the communication cost is also comparable to that of the original FETI method.

A more careful comparison shows that the communication cost when applying the system matrix $\mathcal{A}_r$ to a vector can be implemented such that it is essentially the same as when applying F in standard FETI-DP. For $\mathcal{A}_r$ we have, each, one gather and one scatter operation for λ and for u_Π, and an application of a (parallel) operator to u_Π. Also, the significant computational cost in applying the system matrix $\mathcal{A}_r$ to a vector is the same as in applying F except for the matrix-vector product with $\widetilde{S}_{\Pi\Pi}^{-1}$, which is now shifted to the preconditioner and replaced by $\widehat{S}_{\Pi\Pi}^{-1}$. Although $\widehat{S}_{\Pi\Pi}^{-1}$ appears in two blocks of the preconditioner, in the implementation, the product with a vector has to be carried out only once in each iteration. The same is true for the M-block.

We repeat that in the methods, when using exact subdomain solvers or when using the factorization (4.6), the matrix-matrix product $K_{BB}^{-1}\widetilde{K}_{\Pi B}^T$ is built explicitly in a preprocessing step, as is done generally in standard, exact FETI-DP.

Surprisingly, communication cost for applying the full matrix $\mathcal{A}$ to a vector is also essentially the same as when applying F. However, the computational effort for the methods simultaneously iterating on displacement variables u and Lagrange multipliers λ, is higher because of the higher computational cost in the inner products of the Krylov subspace method. The inner products $u_B^T v_B$ are perfectly parallel and can be calculated by each processor separately but the scalar results of $u_B^{(i)T} v_B^{(i)}$ have to be communicated among the processors.

4.6 Some Technical Remarks

The calculations in Tables 4.1–4.9 were carried out on our 16 processor (8 dual Opteron 248, 2.2 GHz, 8 GB of memory per node) Linux cluster in Essen. Some of the calculations were carried out in parallel using MPICH [60, 61, 62] and PETSc, see Balay et al. [9, 8, 10]. The algebraic multigrid solver SAMG always ran in parallel, using two threads.

In general, we use a sequential direct solver in our FETI-DP implementations for the factorization of $\widetilde{S}_{\Pi\Pi}$ as well as for the factorizations of the subdomain problems. In the 64 bit FETI-DP application used here, for this,

the process calls the direct solver UMFPACK 4.3 [27]. As UMFPACK then is faster, we use it in 32 bit integer and 32 bit pointer mode. So even though the application processes are not subject to the 32 bit memory limit, every single direct solver instance cannot traverse the 32 bit (4 GB) address limit.

4.7 Algebraic Multigrid (AMG)

Multigrid (MG) solvers are iterative subspace correction schemes where the nested subspaces are generally defined by a hierarchy of grids. Multigrid methods have provable optimal complexity for discretizations of a multitude of PDE problems. For an introduction to multigrid algorithms, see, e.g., [137] and many others. A grid hierarchy may readily be available from iterative refinement. Still, difficulties in providing a grid hierarchy have lead to the development of Algebraic Multigrid Methods (AMG), e.g. [112]. In algebraic multigrid methods, the multilevel hierarchy is built based on information of the matrix only. Sometimes some additional information is being used as in [67, 142, 21]. Significant effort has been made to develop parallel scalable Algebraic Multigrid Solvers, e.g. [66, 1].

4.8 Numerical Results

In this section, we present numerical results for the preconditioners analyzed in the previous sections. We apply the preconditioners to 2D and 3D linear elasticity problems. In the tables of this section we denote the iterative substructuring method using the preconditioner $\widehat{\mathcal{B}}_L^{-1}$ for the system (4.1) by inexact FETI-DP or iFETI-DP. The method using the preconditioner $\widehat{\widetilde{\mathcal{B}}}_{r,L}^{-1}$ for the system (2.2) iterating on the variables $[\,\tilde{u}_\Pi^T, \lambda^T\,]^T$ is denoted as the inexact reduced FETI-DP or irFETI-DP. We always state the Krylov subspace method which is used as an accelerator, either GMRES or CG, and also which part of the preconditioner is solved inexactly and by which method. We generally use left preconditioning with GMRES so that we can use the same implementation for the preconditioner as for CG.

4.8.1 Direct Solvers

We use the modified FETI-DP formulations for structured benchmark problems in 2D and 3D using exact solvers (Cholesky or LU decomposition) for the coarse grid problem and the local subdomain problems to verify that the methods perform well in the best case. We use GMRES with left preconditioning and CG to solve the preconditioned problem. In order to fulfill

assumption (4.10), we scale the results obtained from the direct solver. For irFETI-DP and CG, we use $\widehat{S}_{\Pi\Pi}^{-1} = 0.9999^{-2} \cdot \widetilde{S}_{\Pi\Pi}^{-1}$. For iFETI-DP and CG, we use the factorization (4.6) and the scaling $\widehat{S}_{\Pi\Pi}^{-1} = 0.99999^{-2} \cdot \widetilde{S}_{\Pi\Pi}^{-1}$ and $\widehat{K}_{BB}^{-1} = 0.99999^{-2} \cdot K_{BB}^{-1}$.

For the 2D compressible elasticity problems in Table 4.1, we have chosen a larger coarse problem than necessary for scalability. It is well known that vertex constraints are sufficient for this case to ensure numerical scalability, see also Chapter 2.

We see from Table 4.1 that the algorithms perform as expected and converge independently of the number of subdomains. The iteration count and the estimated eigenvalues are almost identical to the numbers which we get from the original FETI-DP method.

Throughout this work eigenvalue estimates are obtained from the Lanczos process in the conjugate gradient method. Here, for the comparison of the different FETI-DP methods we note that the accuracy of these estimates is much lower than the number of given digits suggests. Nevertheless, we can see that the considered FETI-DP algorithms lead to very similar Lanczos eigenvalue estimates.

Direct Solvers							
	2D				3D		
N	64	256	1 024	4 096	64	512	4 096
1/H	8	16	32	64	4	8	16
H/h	8	8	8	8	4	4	4
$\dim(\widetilde{S}_{\Pi\Pi})$	322	1 410	5 890	24 066	324	3 528	32 400
FETI-DP							
CG							
It.	**10**	**11**	**11**	**10**	**14**	**15**	**15**
$\lambda_{\min}$	1.008	1.011	1.007	1.007	1.029	1.026	1.022
$\lambda_{\max}$	2.219	2.344	2.348	2.342	4.107	4.064	4.062
iFETI-DP							
GMRES							
It.	**8**	**8**	**7**	**6**	**14**	**13**	**12**
CG (sc.)							
It.	**10**	**10**	**9**	**9**	**15**	**14**	**13**
$\lambda_{\min}$	1.001	1.001	1.004	1.002	1.020	1.025	1.026
$\lambda_{\max}$	2.219	2.341	2.342	2.341	4.107	4.065	4.063
irFETI-DP							
GMRES							
It.	**8**	**8**	**7**	**6**	**14**	**13**	**12**
CG (sc.)							
It.	**10**	**10**	**9**	**9**	**14**	**13**	**12**
$\lambda_{\min}$	1.001	1.004	1.001	1.001	1.029	1.029	1.030
$\lambda_{\max}$	2.220	2.341	2.344	2.333	4.107	4.066	4.064

Table 4.1

Exact solvers: Comparison of standard FETI-DP with the inexact variants, denoted inexact FETI-DP (iFETI-DP) and inexact FETI-DP on the reduced system (irFETI-DP). Here we use direct solvers for the local subdomain problems and the coarse grid problem. The GMRES iteration count is given for left preconditioning. For the CG accelerated method we use the scaling (irFETI-DP: $\widehat{S}^{-1}_{\Pi\Pi} = 0.9999^{-2} \cdot \widetilde{S}^{-1}_{\Pi\Pi}$; iFETI-DP: $\widehat{S}^{-1}_{\Pi\Pi} = 0.99999^{-2} \cdot \widetilde{S}^{-1}_{\Pi\Pi}$, $\widehat{K}^{-1}_{BB} = 0.99999^{-2} \cdot K^{-1}_{BB}$). The dual Schur complement F is always preconditioned by the Dirichlet preconditioner M_D^{-1}.
— 2D linear elasticity on the unit square for $N = 64$ to $N = 4\,096$ subdomains, Q1-elements, $E = 1$, $\nu = 0.4$, GMRES restart: 50 iterations, Algorithm B (vertex and edge averages), relative residual reduction of 10^{-7}.
— 3D linear elasticity on the unit cube for $N = 64$ to $N = 4\,096$ subdomains, P1-elements, $E = 210$, $\nu = 0.29$, GMRES restart: 50 iterations, Algorithm D_E (only edge averages, no vertices), relative residual reduction of 10^{-7}.

4.8.2 Inexact Coarse Problem

We first investigate the effect of an approximate solver for the coarse grid problem on the iteration count and the estimated eigenvalues of the preconditioned operators.

We need to provide a good preconditioner for $\widetilde{S}_{\Pi\Pi}$ since the condition number of $\widetilde{S}_{\Pi\Pi}$ will grow quickly with the number of subdomains. In Chapter 3 we have investigated how the choice of the coarse problem affects the condition number of the FETI-DP system but we have not yet considered the condition number of the coarse matrix $\widetilde{S}_{\Pi\Pi}$ as we always have used direct solvers.

From Table 4.2 and Figure 4.2 we see that for Algorithm D_E as well as for Algorithm A the condition number seems to be proportional to $1/H^2$. However, from Table 4.3 we see that the condition number of the coarse problem does change only weakly for different subdomain sizes if the number of subdomains is kept fixed.

	$\kappa(\widetilde{S}_{\Pi\Pi})$								
1/H	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Alg. D_E	15.6	76.2	173	313	488	704	955	1 246	1 573
Alg. A	10.7	60.0	127	225	337	477	628	806	996
Alg. D_F	90	337	686	1 154	1 733	2 424	3 226	4 139	5 163

Table 4.2
Condition number of $\widetilde{S}_{\Pi\Pi}$ for Algorithm D_E, Algorithm A, and Algorithm D_F for 3D linear elasticity, $H/h = 5$.

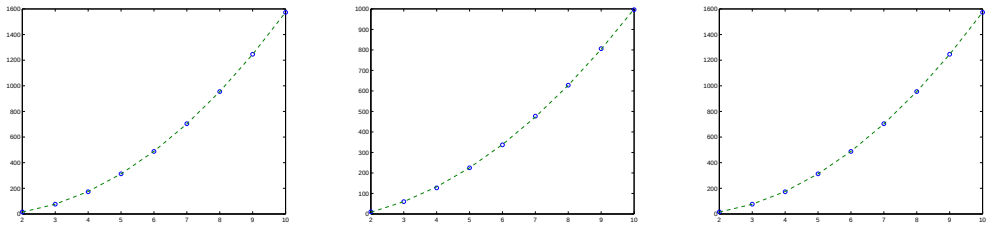

Figure 4.2
Circles: Condition number of $\widetilde{S}_{\Pi\Pi}$ for Algorithm D_E, Algorithm A and Algorithm D_F plotted versus $1/H$, $H/h = 5$; 3D linear elasticity. Dashed line: Fit of a second order polynomial.

	$\kappa(\widetilde{S}_{\Pi\Pi})$							
H/h	2	4	6	8	10	12	14	16
Alg. D_E	170	164	159	156	151	150	148	148
Alg. A	123	119	118	117	116	116	115	115

Table 4.3
The condition number of $\widetilde{S}_{\Pi\Pi}$ for a fixed number of subdomains, $N = 4 \times 4 \times 4 = 64$.

We use cycles of the parallel algebraic multigrid package SAMG [131, 130, 129, 137] by Stüben and Clees to precondition $\widetilde{S}_{\Pi\Pi}$. SAMG is extremely memory efficient and the version we applied uses shared memory parallelism. In the experiments in Table 4.4 we use V-cycles and ILU(0)-smoothing within SAMG. We see that the method performs remarkably well in 2D and 3D for the structured benchmark problems. The iteration count is comparable to that of the original FETI-DP method, cf. Table 4.1, and the estimated condition number is only slightly higher. By increasing the number of SAMG cycles we can recover the eigenvalues of the original FETI-DP method.

Whenever scaling is used in the experiments, it is calculated using three digits of an eigenvalue estimate obtained from a few, typically less than ten, CG iterations with $\tilde{S}_{\Pi\Pi}$ preconditioned by SAMG. In the tables these cases are captioned by "sc.". From the experiments we do not see the necessity of scaling when using GMRES.

In Table 4.5 we present a larger test problem with 13 824 subdomains, which we are unable to solve with our current FETI-DP implementation since the direct factorization of $\widetilde{S}_{\Pi\Pi}$ requires too much memory (> 2 GB); see Section 4.6 for further details.

Next, we consider a larger and unstructured mesh. The mechanical part shown in Figure 4.3 is discretized using 1 291 933 linear, tetrahedral elements. The resulting problem has 841 836 degrees of freedom; see Chapter 3. Using ParMetis [70], we partition the mechanical part into 1 024 and 2 048 very small subdomains in order to obtain a coarse problem of reasonable size.

Typically, it is more efficient to partition this mechanical part into a considerably smaller number of subdomains; see [76]. From Table 4.8 we see that even for this industrial benchmark problem the GMRES iteration count remains acceptable and compares well with standard FETI-DP using CG.

	Inexact solver for the coarse problem SAMG						
	2D				3D		
N	64	256	1 024	4 096	64	512	4 096
1/H	8	16	32	64	4	8	16
H/h	8	8	8	8	4	4	4
$\dim(\widetilde{S}_{\Pi\Pi})$	322	1 410	5 890	24 066	324	3 528	32 400
iFETI-DP							
GMRES							
It.	**8**	**8**	**8**	**7**	**14**	**13**	**12**
irFETI-DP							
GMRES							
It.	**8**	**8**	**7**	**7**	**14**	**13**	**12**
GMRES (sc.)							
It.	**8**	**8**	**7**	**7**	**14**	**14**	**13**
CG (sc.)							
It.	**9**	**9**	**9**	**10**	**14**	**15**	**15**
$\lambda_{\min}$	1.013	1.021	1.035	1.037	1.032	1.029	1.028
$\lambda_{\max}$	2.258	2.419	2.472	2.545	4.276	4.523	4.864

Table 4.4

Inexact solver for the coarse grid problem: Performance of the inexact variants of FETI-DP, denoted inexact FETI-DP (iFETI-DP) and inexact FETI-DP on the reduced system (irFETI-DP). We use direct solvers for the local problems and SAMG $\widetilde{S}_{\Pi\Pi}$ using two V-cycles for each outer Krylov subspace iteration. The V-cycles use one sweep of ILU(0) as pre- and post-smoother. For some of the calculations we use scaling ("sc.").
— 2D linear elasticity on the unit square for $N = 64$ to $N = 4\,096$ subdomains, Q1-elements, $E = 1$, $\nu = 0.4$, GMRES restart: 50 iterations, Algorithm B (vertex and edge averages), relative residual reduction of 10^{-7}.
— 3D linear elasticity on the unit cube for $N = 64$ to $N = 4\,096$ subdomains, P1-elements, $E = 210$, $\nu = 0.29$, GMRES restart: 50 iterations, Algorithm D_E (only edge averages, no vertices), relative residual reduction of 10^{-7}.

Inexact solver for the coarse problem (3D)								
				irFETI-DP / SAMG				
				GMRES	GMRES (sc.)	CG (sc.)		
N	1/H	H/h	$\dim(\widetilde{S}_{\Pi\Pi})$	It.	It.	It.	$\lambda_{\min}$	$\lambda_{\max}$
13 824	24	4	114 264	**12**	**12**	**14**	1.030	4.731

Table 4.5
A problem with a large number of subdomains: Inexact FETI-DP on the reduced system (irFETI-DP). Exact solvers for the local problem and SAMG [131, 130, 129, 137] using two V-cycles per outer Krylov subspace iteration to precondition $\widetilde{S}_{\Pi\Pi}$. The V-cycles use one sweep of ILU(0) as pre- and post-smoother; scaling was used for two of the calculations ("sc.").
— 3D linear elasticity on the unit cube, $N = 13\,824$ ($24 \times 24 \times 24$) subdomains, 192 ($4^3 \times 3$) d.o.f. per subdomain, 1 167 051 total d.o.f., P1-elements, $E = 210$, $\nu = 0.29$, GMRES restart: 50 iterations, Algorithm D_E (edge averages), relative residual reduction of 10^{-7}.

4.8.3 Inexact Neumann Problems

Here, we only present preliminary results for the inexact solution of the local Neumann problems using factorization (4.6). For a first set of experiments, presented in Table 4.6, we use incomplete Cholesky decompositions with a threshold of 10^{-4} for a structured 3D elasticity problem. We use a renumbering [27] before the incomplete factorization. For these calculations the incomplete Cholesky factorization (ICC) uses on average about 50 percent of the memory required for an exact factorization. We only present results for GMRES. From these results we see that the GMRES iteration count is still comparable to standard FETI-DP using CG. Here, we do not use any scaling to satisfy (4.10), and we also note that ICC is not an optimal preconditioner.

In another set of experiments, presented in Table 4.7, we consider a larger number of smaller subdomains in 2D and use incomplete Cholesky decompositions for the local Neumann problems with a threshold of 10^{-2}. The local subdomain sizes remain fixed. In these experiments the incomplete Cholesky factorizations use less than 65 percent of the memory required for the exact factorizations. We see that the number of iterations remains bounded as the number of subdomains increases from 16 to 1 024, as is expected.

Inexact solvers for the local problems 3D, ICC(1e-4)			
	N	8	64
	1/H	2	4
	H/h	14	14
FETI-DP			
CG			
	It.	**18**	**23**
	$\lambda_{\min}$	1.03	1.03
	$\lambda_{\max}$	12.94	10.45
iFETI-DP			
GMRES			
	It.	**22**	**22**

Table 4.6

Inexact solver for the local Neumann problems: Comparison of standard, exact FETI-DP with inexact FETI-DP (iFETI-DP) using incomplete Cholesky (ICC) with a threshold of 10^{-4} *for the local Neumann problems and a direct solver for the coarse grid problem. In these experiments the incomplete Cholesky factorization uses about half of the memory that is required for the total factorization.*

— 3D linear elasticity on the unit cube for $N = 8$ *and* $N = 64$ *subdomains,* $8\,232$ *(*$=14^3 \times 3$*) d.o.f. per subdomain,* $59\,049$ *and* $446\,631$ *total d.o.f., respectively; P1-elements,* $E = 210$, $\nu = 0.29$, *GMRES restart:* 50 *iterations, Algorithm* D_E *(edge averages), relative residual reduction of* 10^{-7}.

	Inexact solvers for the local problems 2D, ICC(1e-2)							
N	16	36	64	100	256	400	768	1 024
iFETI-DP GMRES								
It.	14	14	13	13	12	12	11	11

Table 4.7

Inexact solver for the local Neumann problems: Comparison of standard FETI-DP with inexact FETI-DP (iFETI-DP) using incomplete Cholesky (ICC) with a threshold of 10^{-2} *for the local Neumann problems and a direct solver for the coarse grid problem. In these experiments, the incomplete Cholesky factorization uses less than two thirds of the memory that is required for the total factorization.*
— 2D linear elasticity on the unit cube for $N = 4$ *to* $N = 1\,024$ *subdomains,* $128\ (= 8^2 \times 2)$ *d.o.f. per subdomain, Q1-elements,* $E = 1$, $\nu = 0.4$, *GMRES restart: 50 iterations, Algorithm B (vertices and edge averages), relative residual reduction of* 10^{-7}.

Figure 4.3

Mechanical part C courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany.

	Mechanical Part	
N	1 024	2 048
FETI-DP		
CG		
It.	**41**	**48**
$\lambda_{\min}$	1.04	1.04
$\lambda_{\max}$	33.31	44.63
irFETI-DP / SAMG		
GMRES		
It.	**53**	**64**

Table 4.8
Mechanical part, see Figure 4.3, 841 836 d.o.f., partitioned into 1 024 *and* 2 048 *subdomains. The coarse problem has a size of* 10 380 *and* 19 515 *d.o.f., respectively. We use* 4 *V-cycles of SAMG [131, 130, 129, 137] in each outer Krylov subspace iteration. Two Gauss-Seidel sweeps as pre- and two as post-smoother are used. The Gauss-Seidel sweeps use CF-ordering for pre-smoothing and the reverse for post-smoothing.*
— 3D linear elasticity on the unit cube for $N = 1\,024$ *and* $N = 2\,048$ *subdomains, P1-elements,* $E = 210$, $\nu = 0.29$, *GMRES restart:* 100 *iterations, Algorithm* D_E *(edge averages), relative residual reduction of* 10^{-7}.

4.8.4 Inexact Neumann, Dirichlet, and Coarse Problems

In this section we present some results for iFETI-DP where we apply a preconditioner directly to $\widetilde{K}$ without using (4.6). This refers to an algorithm where the local subdomain Neumann solves are inexact as well as the coarse problem. In a next step, we also introduce an inexact Dirichlet preconditioner

$$\widehat{M}^{-1}_{\mathrm{FETI}_D} := B_{B,D}(R^B_\Delta)^T(K_{\Delta\Delta} - K_{\Delta I}\widehat{K}^{-1}_{II}K^T_{\Delta I})R^B_\Delta B^T_{B,D}. \tag{4.12}$$

For all of these experiments we use the algebraic multigrid code (BoomerAMG), cf. Section 4.9. As all eigenvalues of the AMG operator are smaller than one, the inexact Dirichlet preconditioner is still positive definite. Note that the $(1 + \log(H/h))^2$ bound presumes the use of the exact Dirichlet preconditioner. The discretizations are from spectral elements, see Chapter 7. The columns are captioned in the following way.

The caption (e/e/e) refers to exact standard FETI-DP using the exact Dirichlet preconditioner. Here, also condition numbers are given as CG is used. The caption (i/e/e) refers to iFETI-DP using the factorization (4.6) to provide an inexact solver for the subdomain Neumann problems but an exact solver for the coarse problem and using the exact Dirichlet preconditioner. The caption (i/i/e) then refers to directly applying the inexact solver to $\widetilde{K}$ without using the factorization (4.6) but applying the exact Dirichlet preconditioner. Then in a last step, (i/i/i), everything is solved inexactly, i.e. we also apply the inexact Dirichlet preconditioner.

From the results in Table 4.9 we see that all methods are scalable with respect to the numbers of subdomains, and the performance of the method that uses all inexact solvers is still astonishingly competitive with standard FETI-DP. In fact, the iteration count increases noticeably once the subdomain Neumann problems are solved inexactly. But then, the inexact solution of the coarse problem and the use of the inexact Dirichlet preconditioner do not change the iteration count anymore. Note that we only have a small coarse problem here.

For some results of irFETI-DP applied to higher order discretizations, see Chapter 7.

p	N	FETI-DP	$\lambda_{\min}$	$\lambda_{\max}$
32	6	6	3.42	1.0000
	16	17	9.48	1.0012
	24	25	10.57	1.0012
	24	24	10.69	1.0018
	24	25	10.75	1.0016

		iFETI-DP			
p	N	It. (e/e/e)	It. (i/e/e)	It. (i/i/e)	It. (i/i/i)
32	4	6	13	13	13
	16	16	20	21	22
	64	24	29	30	30
	100	24	30	30	30
	144	24	30	29	30

Table 4.9

Spectral elements of fixed polynomial degree ($p = 32$), fixed subdomain sizes ($H/h = 1$), increasing number of subdomains, $\rho = 1$, random right hand side, rtol=10^{-7}. Inexact FETI-DP for the block matrices using BoomerAMG and GMRES, Neumann problems/coarse problem/Dirichlet problems: (in)exact/(in)exact/(in)exact.

4.9 Parallel Results

In this section, we focus on the inexact reduced method irFETI-DP; see formulation (4.3) and Section 4.4.2, respectively. Here, we use exact solvers for the subdomain problems but are able to use inexact solvers for the global coarse problem. In order to obtain numerical and parallel scalability for our distributed memory FETI-DP implementation even for a large number of subdomains, a good parallel preconditioner for the coarse problem is essential. For our parallel computations in this section, we use BoomerAMG [66] by Henson and Meier Yang to precondition the FETI-DP coarse problem. BoomerAMG is a highly scalable distributed memory parallel algebraic multigrid solver and preconditioner; it is part of the high performance preconditioner library hypre [35, 36, 37].

The purpose of this section is to validate that the parallel performance of the inexact FETI-DP algorithm is competitive with the standard FETI-DP method when the number of subdomains is not too large. Furthermore, we present parallel results for irFETI-DP in the case of a large number of subdomains when a direct solution of the coarse problem is not feasible anymore.

In Tables 4.10 and 4.11, our standard FETI-DP implementation [76, 77] using CG is compared to the new implementation of the inexact reduced irFETI-DP method using GMRES. The results were obtained on the MCR cluster at the Lawrence Livermore National Laboratory, Livermore, CA, USA. The MCR cluster has 2 304 Intel Xeon 2.4 GHz processors, organized as dual processor nodes with 4 GB of memory for each node, and it uses a Quadrics interconnect. The implementation of standard FETI-DP uses the direct sparse solver MUMPS 4.5.0 [4, 5, 6] to solve the coarse problem. This direct sparse solver can also be used in parallel mode to solve the coarse problem, but there are approaches that are more appropriate for this situation, e.g. an explicitly calculated, distributed inverse as used in [13] or the approach from [141]. In all of our experiments we have used MUMPS in sequential mode to solve the coarse problem.

In the implementation of irFETI-DP, we employ one iteration of BoomerAMG to precondition the coarse problem. Both implementations use MPI and PETSc [9, 8, 10], and both use the direct sparse solver [4, 5, 6] as the local subdomain solver.

In Table 4.10 we present weak scaling results for the algorithms on the MCR cluster. The model problem is 3D linear elasticity with Young's modulus $E = 210$ and Poisson's ratio $\nu = 0.29$. We consider the unit cube decomposed into cubic subdomains. As in [76] and [77], we use edge average constraints to form the coarse problem. It can be seen that our implementations of irFETI-DP and standard FETI-DP show nearly the same parallel

				FETI-DP		irFETI-DP	
N	**D.o.f.**	**Coarse**	**Proc.**	**It.**	**Time**	**It.**	**Time**
64	1 369 599	324	16	25	**144s**	28	**148s**
512	10 744 731	3 528	128	27	**165s**	27	**172s**
1 728	36 026 967	13 068	432	27	**233s**	28	**194s**

Table 4.10
Results for standard FETI-DP and the inexact reduced variant (irFETI-DP) obtained on LLNL's MCR cluster, ranked 34th among the worlds fastest computers (see `www.top500.org`*) at the time of the experiments; 3D linear elasticity benchmark problem, unit cube.*

performance; the timings and the iteration counts differ only slightly. We can see that the irFETI-DP is scalable up to 1 728 subdomains and 36 million unknowns. We note that we have not done any extensive testing of parameters for BoomerAMG. In our experiments, irFETI-DP seems to scale slightly better than standard FETI-DP; cf. Table 4.10.

				FETI-DP		irFETI-DP	
N	**D.o.f.**	**Coarse**	**Proc.**	**It.**	**Time**	**It.**	**Time**
4 096	352 947	32 400	16	14	**218s**	21	**14s**
27 000	2 260 713	227 070	120	failed	failed	22	**38s**

Table 4.11
Large number of subdomains. Results for standard FETI-DP and the inexact reduced variant (irFETI-DP) obtained on LLNL's MCR cluster; 3D linear elasticity benchmark problem, unit cube.

In Table 4.11, we provide results for large numbers of smaller subdomains in order to gain insight into potential scalability to a very large number of processors. We see that already for the case of 4 096 subdomains the inexact method is much faster. Furthermore, the inexact method is able to cope with as many as 27 000 subdomains whereas the exact method is not. We also expect that the scalability of irFETI-DP will improve once larger subdomain problems are used. The results give rise to the hope that scalability to more than tens of thousands of processors might be possible with this method and this implementation.

4.10 Conclusions

We have seen that for a large number of subdomains or processors, the FETI-DP coarse problem can become very large. This may lead to a bottleneck when solved with a direct method. For problems of such a size, there is a need to solve the coarse problem inexactly to maintain parallel scalability. An inexact solution of the subdomain problems may also be desirable.

In this chapter, we presented a framework for a family of inexact FETI-DP methods, together with a convergence theory and numerical results. The framework allows for the construction of different algorithms with approximative solvers for the coarse problem as well as for the subdomain problems.

In our numerical experiments, we observed that the convergence rates of the inexact methods are comparable to those of the standard FETI-DP method if good approximative solvers are used as preconditioning blocks.

Parallel experiments were carried out for the inexact reduced method, irFETI-DP, where the subdomain problems are still solved exactly, and only the coarse problem is solved inexactly using an algebraic multigrid preconditioner. The new method inherits the numerical and parallel scalability of the original FETI-DP method and is also applicable when the coarse problem size becomes too large to be solved by a direct factorization method. Using this approach, we hope to maintain parallel scalability up to a very large number of processors.

Chapter 5

Heterogeneous Elasticity Problems

Domain decomposition methods provide a good approach to obtaining robust and parallel scalable solvers. In the present chapter, we will focus on heterogeneous elasticity problems with large discontinuities in the material stiffnesses. It was shown in [84] that selecting certain edge averages and first order moments and, in some very special cases, certain vertices as primal constraints, yields a robust condition number estimate. The central assumption is that on every face the rigid body modes can be controlled and, additionally, almost uniform bounds on some related functionals are satisfied. We provide results which confirm the theoretical findings in [84] and show that, in some cases, first order moments are indeed necessary to obtain a good convergence rate. In the theory it is assumed that the edges of the subdomains are straight and that large material discontinuities are aligned with the subdomain boundaries. We relax these assumptions and provide numerical results for decompositions with curved edges and with material discontinuities which are not aligned with the interface. The following presentation is based on Klawonn and Rheinbach [77].

We expect that our numerical results also provide insight into the performance of the more recently developed Neumann-Neumann methods with constraints, known as the BDDC algorithms, cf. [29, 97, 98, 92], since Mandel, Dohrmann, and Tezaur [98] have shown that, for any given set of constraints, the BDDC and FETI-DP methods have almost all of their eigenvalues in common; see also Li and Widlund [92] for an alternative proof, see also Section 2.4.

5.1 Additional Constraints

The simplest choice of primal variables for FETI-DP and BDDC is to select them as certain primal vertices of the subdomains, see Chapter 2. Unfortunately, this choice does not always lead to good convergence results in three dimensions, see the beginning of Chapter 3. Therefore, we have discussed average constraints over edges to form the coarse problem, see Chapter 3. To obtain robust condition number bounds for highly heterogeneous materials in all cases of material distributions, additional constraints have to be introduced. In the case of heterogeneous materials, we need to introduce the notion of edge first order moments, see [84], in addition to the edge averages that we have used in Chapter 3 for homogeneous elasticity.

The rigid body modes $r_1, \ldots, r_6$, restricted to a straight edge provide only five linearly independent vectors, since one rotation is always linearly dependent on other rigid body modes. For the following definition, we assume that we have used an appropriate change of coordinates such that the edge under consideration coincides with the x_1-axis, and the special rotation is then r_6. The edge averages and first order moments over this specific edge $\mathcal{E}$ are of the form

$$\frac{\int_{\mathcal{E}} r_k^T u dx}{\int_{\mathcal{E}} r_k^T r_k dx}, \quad k \in \{1, \ldots, 5\}, \tag{5.1}$$

where $u = (u_1^T, u_2^T, u_3^T)^T \in W_\Gamma^h$. We note that for edges which are not straight we can use all six rigid body modes to construct three average and three first order moment constraints.

From the inner products of u_E with the translational and the rotational rigid body modes, we obtain three averages and either two or three first order moments, see (5.1). On a straight edge the three averages and three rotations are linearly dependent, and one of the rotations can be discarded.

Using a transformation of basis, see Chapter 3, the averages and moments are explicitly introduced as new variables into the basis of our finite element space. These variables then form a part of the set of primal displacement variables. In this manner we force our edge averages and first order moments to be the same across the interface.

If at least one edge average of one displacement component is imposed as a constraint on an edge, we will denote this edge as *primal*. At most three average constraints can be imposed on such an edge. Furthermore, if three edge averages and additionally two or three edge first order moments are imposed as primal constraints on an edge we will denoted this edge as *fully primal*.

We now describe how the transformation matrix for such a change of

basis can be constructed. Here, we restrict ourselves to the construction of the basis transformation for a single, fully primal edge; see Section 3.3 for a detailed, algorithmic description.

We consider the six rigid body modes r_i, $i = 1, \ldots, 6$; cf. Section 3.1. Next, we orthogonalize the rigid body modes on the edge against each other using a stable formulation of the Gram-Schmidt process, e.g. modified Gram-Schmidt. We note that the translational rigid body modes are already orthogonal to each other and thus, we only have to start with the rotations in order to obtain an orthogonal basis of rigid body modes on the edge $\mathcal{E}$. We denote the orthogonal basis obtained by this process by $(\hat{r}_j)_{j=1,\ldots,\ell}$, with $\ell \in \{5, 6\}$. When restricted to a straight edge $\mathcal{E}$, one of the rotations is linearly dependent on the others and should vanish when modified Gram-Schmidt is used; cf. also the discussion at the end of Section 3.1. Then, we only have a five dimensional basis.

Let us assume that the vector of nodal unknowns u_E has length n. We then consider the set of vectors $\{(\hat{r}_j)_{j=1,\ldots,\ell}, (e_i)_{i=1,\ldots,n}\}$, where e_i is the unit vector with one at the ith component and zero otherwise, which is associated with the ith d.o.f. on the fully primal edge. Starting with the orthogonalized rigid body modes $(\hat{r}_j)_{j=1,\ldots,\ell}$, we orthogonalize and normalize the set of $n + \ell$ vectors, using modified Gram-Schmidt. We discard the ℓ linearly dependent vectors and use the remaining n orthogonal vectors to define the column vectors of our transformation matrix T_E.

The transformation matrix T_E performs the desired change of basis from the new basis to the original nodal basis. From our construction, T_E is an orthogonal matrix. Denoting the edge unknowns in the new basis by $\overline{u}_E$, we have

$$u_E = T_E \overline{u}_E.$$

In the case where only averages are used as primal constraints we have explicitly set up the matrix for the basis transformation; see Chapter 3. For this case we can also apply the construction used here, i.e. we orthogonalize only against the translational rigid body modes. Only edge averages are then introduced as new variables, and the remaining basis functions will have zero edge average.

In the theory presented in Klawonn and Widlund [84], it is assumed that the subdomains are polytopes with good aspect ratios and that the edges are straight. Furthermore, large material discontinuities should be aligned with the interface. For our FETI-DP algorithm, using a well selected set of primal constraints of edge averages or edge first order moments and in some special cases also primal vertices, we have Theorem 5.1.1, cf. [84]; see also Theorem 3.3.1 (p. 67) for the homogeneous case.

Theorem 5.1.1 *The condition number satisfies*

$$\kappa(M^{-1}F) \leq C\,(1+\log(H/h))^2.$$

Here, $C > 0$ is independent of h, H, and the values of the coefficients G_i.

Let us give some assumptions from [84] which are sufficient to obtain a condition number estimate as in Theorem 5.1.1. In [84, Section 5], the definition of a *fully primal face* is introduced as a face which has six edge constraints, averages or first order moments, on edges which belong to the boundary of that face, such that the rigid body modes on that face are controlled. The set of constraints has to control the rigid body modes in the sense that if all functionals representing the constraints, cf. (5.1), vanish for a rigid body mode, then the rigid body mode must vanish itself.

A sufficient condition for Theorem 5.1.1, cf. [84, Section 8.2], is to assume that every face is fully primal, every edge which belongs to more than three subdomains is fully primal, and every vertex is primal. In fact, not every such edge and every face has to be fully primal and not every vertex needs to be primal as long as for every pair of subdomains $\{\Omega_i, \Omega_j\}$ which has a (non fully primal) face or edge or a (nonprimal) vertex in common there is a path connecting them. This path has to lead, possibly through several other subdomains Ω_k, only through fully primal faces such that the stiffnesses G_k associated with Ω_k are never smaller than the minimum of the stiffnesses associated with Ω_i and Ω_j. Such a path is called an *acceptable path*. For more details on refined path concepts see [84, Section 5].

5.2 Necessary and Sufficient Constraints

In this section, we consider different model problems which numerically confirm the theoretical findings in Klawonn and Widlund [84].

In order to control the rigid body motions of a subregion, we need at least six constraints. In [84, Section 5], two model problems are considered to develop an understanding of the type and number of necessary and sufficient constraints needed in order to obtain a robust and scalable domain decomposition method.

Let us consider two subdomains with a high Young's modulus surrounded by subdomains with a small Young's modulus. This configuration will have six low energy modes related to the six rigid body motions of the union of the two subdomains with high stiffness. In our preconditioner, if we do not introduce any primal constraints, we will have twelve rigid body motions, related to the two subdomains moving individually. Thus, the preconditioner

will be far from spectrally equivalent. Therefore, we need to introduce primal constraints on the face common to the two subdomains with high Young's modulous.

In our model problems, we consider the unit cube in $\mathbb{R}^3$, decomposed into a set of smaller, cubic subdomains with sidelength H. The unit cube is only fixed at one face and a volume force is applied. In all of our experiments, we use four node, tetrahedral finite elements. The Poisson's ratio in our linear elasticity problem is always $\nu = 0.29$, and the Young's modulus will be given separately for the different problems. All computations in this chapter were carried out using PETSc [9, 8, 10], on our 16 processor (2.2 GHz dual Opteron 248; Gigabit Ethernet; 4 GB memory for each processor) computing cluster in Essen. We use UMFPACK 4.3 [27] as a local subdomain and a coarse problem direct solver.

We first consider a decomposition of the unit cube into $3 \times 3 \times 4$ subdomains of 1 536 d.o.f. each, where we have two interior cubic subdomains made of the same material having a face in common and being surrounded by cubic subdomains made of a material with much smaller Young's modulus; cf. Figure 5.1.

Here, we check if six constraints for a face are necessary. We start with making all edges of the decomposition primal, using all three edge averages on each edge, one average for each displacement component. Then, we successively reduce the number of constraints at that common face until no constraints are imposed anymore. Since we always impose three average constraints on a primal edge, we first choose $12 = 4 \times 3$ averages, followed by $9 = 3 \times 3$, $6 = 2 \times 3$, $3 = 1 \times 3$, and 0 averages. From the numerical results presented in Table 5.1, we see that, for this configuration, six linearly independent constraints are necessary to obtain a robust domain decomposition method. We note that in the two cases of two primal edges considered here, we always have one linearly dependent constraint.

We next consider a model problem where two subdomains are again surrounded by subdomains with much smaller stiffnesses, i.e. Young's moduli. Furthermore, we assume that these two special subdomains share only an edge; cf. Figure 5.3. This configuration does not often occur in irregular decompositions where usually more than 99% of the edges only have a multiplicity of three. In [84] it was shown that a well selected set of primal constraints, which has five linearly independent primal constraints related to that special edge shared by the two stiffer subdomains and otherwise six linearly independent edge constraints for each face, is sufficient to prove a condition number bound as in Theorem 5.1.1. In [84], the five linearly independent constraints are chosen as three edge averages and two properly chosen first order moments; cf. also (5.1). The six linearly independent con-

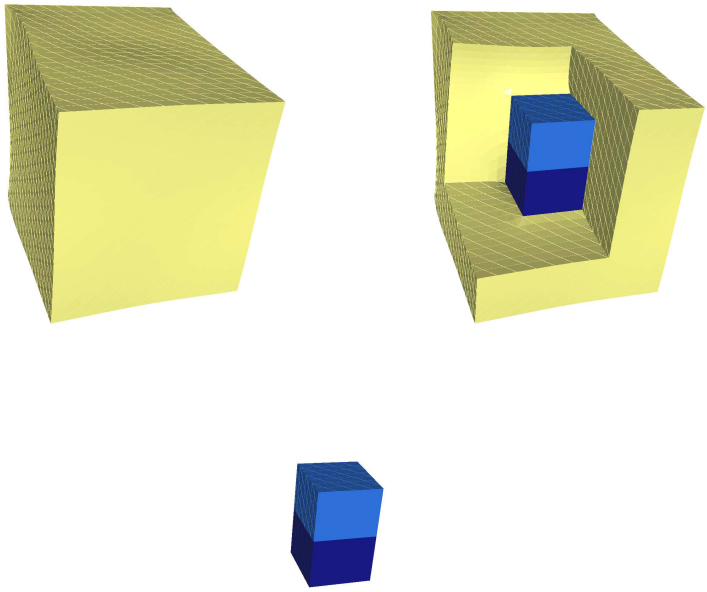

Figure 5.1
Two stiff cubic subdomains sharing a face surrounded by softer material. Left: Unit cube. Right: Unit cube cut open with two stiff subdomains inside. Lower: Two stiff subdomains shown without surrounding softer material.

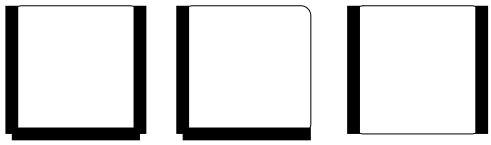

Figure 5.2: *Edges in a U-shaped, L-shaped, and parallel distribution.*

straints for each face can be chosen as edge averages (and moments) over appropriately chosen edges of the considered face. In a set of experiments, we have tested different combinations of edge constraints on the specific edge shared by the two stiffer subdomains; cf. Table 5.2. In the case of three constraints only edge averages are used, in the case of five, additionally two first order moments are applied. On all other edges, an edge average over each displacement component is used to define the primal constraints. We

	Number of primal edges					
E_1/E_2	4	3 (U)	2 (L)	2 (II)	1	0
1	**9.75**	**10.11**	12.77	10.86	13.37	15.67
10^3	**9.24**	**24.28**	1.74×10^3	1.11×10^3	2.16×10^3	4.31×10^3
10^6	**9.24**	**24.64**	1.73×10^6	1.11×10^6	2.14×10^6	4.25×10^6

Table 5.1

Two stiff subdomains sharing a face $\mathcal{F}$. Condition number estimates for different numbers of edge average constraints. The notation U, L, and II denote a U-shaped, an L-shaped, and a parallel distribution of the primal edges of the face $\mathcal{F}$, cf. Figure 5.2. Young's modulus: $E_2 = 210$. Stopping criterion: Relative residual reduction of 10^{-10}. Primal constraints: Edge averages.

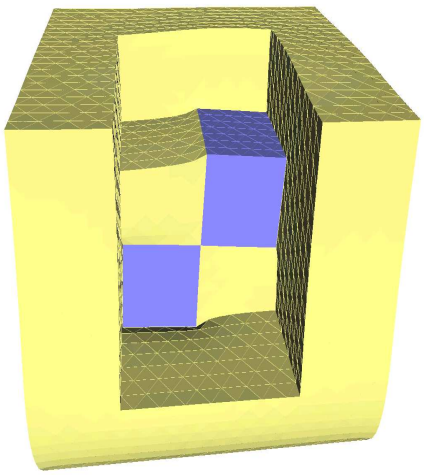

Figure 5.3

Two stiff cubic subdomains sharing an edge surrounded by softer material. Unit cube Ω cut open in front and on top.

see that using no constraints or only edge average constraints on the specific edge leads to a large condition number. Applying all five constraints leads to a good condition number which is bounded independently of the jump in the Young's moduli. It is striking that the iteration counts in Table 5.2 for the case of no constraints or only edge constraints and large coefficient jumps in the Young's moduli are not increased accordingly to the very large condition

numbers. This is due to the effect that the spectrum is still nicely clustered with the exception of a few outliers; see also the theory developed in Klawonn and Widlund [84]. Let us note that not only the condition number but also the iteration count increases dramatically when we have many bad edges instead of only one; see Table 5.3 and Figure 5.6. We have analyzed the spectrum for a single bad edge in more detail numerically for the case of edge constraints with and without additional first order moments.

	−3			+0			+2		
E_1/E_2	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
10^0	29	9.31	1.0111	28	9.20	1.0115	28	**9.19**	1.0113
10^1	31	12.13	1.0105	30	9.14	1.0099	30	**9.14**	1.0098
10^2	36	51.15	1.0115	31	10.61	1.0096	30	**9.11**	1.0094
10^3	47	4.41×10^2	1.0113	37	75.72	1.0081	30	**9.11**	1.0084
10^4	48	4.34×10^3	1.0191	41	7.27×10^2	1.0080	30	**9.10**	1.0080
10^5	65	4.33×10^4	1.0156	48	7.24×10^3	1.0080	30	**9.10**	1.0080
10^6	70	4.33×10^5	1.0215	47	7.24×10^4	1.0116	30	**9.10**	1.0080

Table 5.2

Straight edge: Unit cube decomposed into $3 \times 4 \times 4 = 48$ *brick-shaped subdomains of* 1 536 *d.o.f. each,* 55 506 *total d.o.f., 75 edges, edges use three edge averages. One special edge:* -3*, no constraints on this edge;* 0*, only averages;* $+2$*, averages and, additionally, two first order moments. Stopping criterion: Relative residual reduction of* 10^{-10}*.*

From the theory in Klawonn and Widlund [84], it is expected that for the problem with just one bad edge we only have two large eigenvalues which are outliers in an otherwise nicely clustered spectrum. This also explains the good iteration counts in Table 5.2. To confirm this numerically, we compute all eigenvalues for a small model problem. Here, the unit cube is decomposed into $2 \times 2 \times 2 = 8$ regular cubic subdomains with 1 029 d.o.f. each, resulting in 6 591 global d.o.f. We assume again that we have two stiff subdomains sharing an edge surrounded by softer material; cf. Figure 5.5. The ratio of the different Young's moduli is $E_1/E_2 = 10^6$ with $E_2 = 210$. As constraints we impose three edge averages, one for each displacement component, on each edge but no primal vertices. In this case, we have a large condition number due to two outliers in the spectrum, related to the two rotations which are not controlled due to the missing first order moments; cf. Figure 5.4. If we remove those two outlying eigenvalues from the diagram, we see that the

remaining spectrum is still nicely clustered, cf. the diagram on the right hand side in Figure 5.4. Next, we consider the case when two additional first order moments are imposed on the bad edge shared by the two stiffer subdomains. Here, the spectrum is nicely clustered, and the ratio of the two extreme nonzero eigenvalues is bounded by a small number; cf. Figure 5.5.

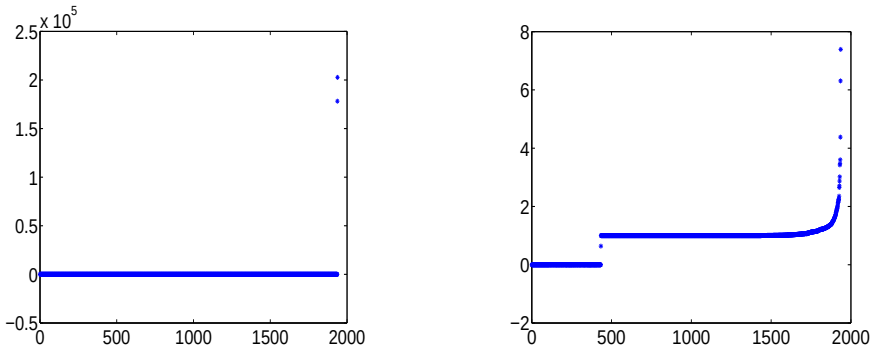

Figure 5.4

Eigenvalues for FETI-DP using only edge averages, no primal vertices. Left: All eigenvalues. Two eigenvalues are very large ($\approx 2 \times 10^5$). Right: All but the two largest eigenvalues.

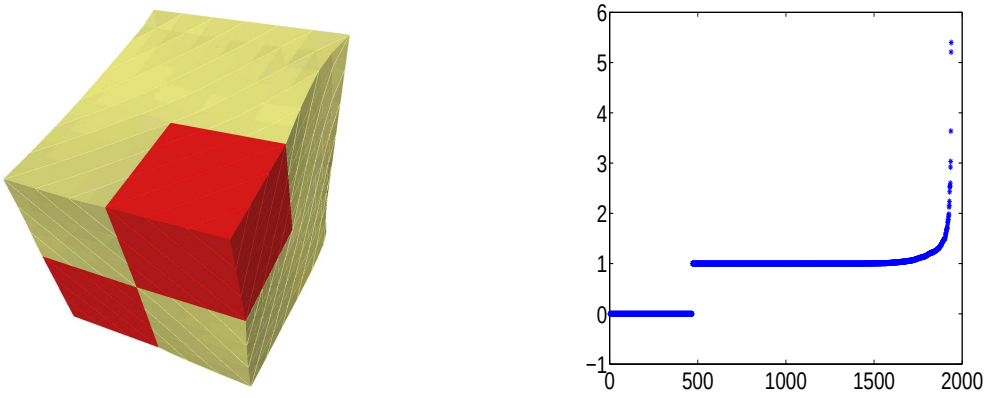

Figure 5.5

Left: Two stiff cubic subdomains sharing a straight edge surrounded by softer material. Right: All eigenvalues for FETI-DP using edge averages and two first order moments on the shared edge, no primal vertices.

Next, we analyze a more involved example with many bad edges, where we will see that additional first order moments not only improve the condition

number but can be absolutely necessary to obtain convergence. We consider a linear elasticity model problem with a material consisting of different layers as shown in Figure 5.6, where the homogeneous layer is made of the softer material. The ratio of the different Young's moduli is $E_1/E_2 = 10^6$ with $E_2 = 210$. Here, in addition to three edge averages on each edge, we have also used two first order moments as primal constraints on all edges. The results in Table 5.3 clearly show that the additional first order moments help to improve the convergence significantly; see [84] for theoretical results. In Table 5.4 the parallel scalability is shown for a cube of eight layers with a material distribution as in Figure 5.6.

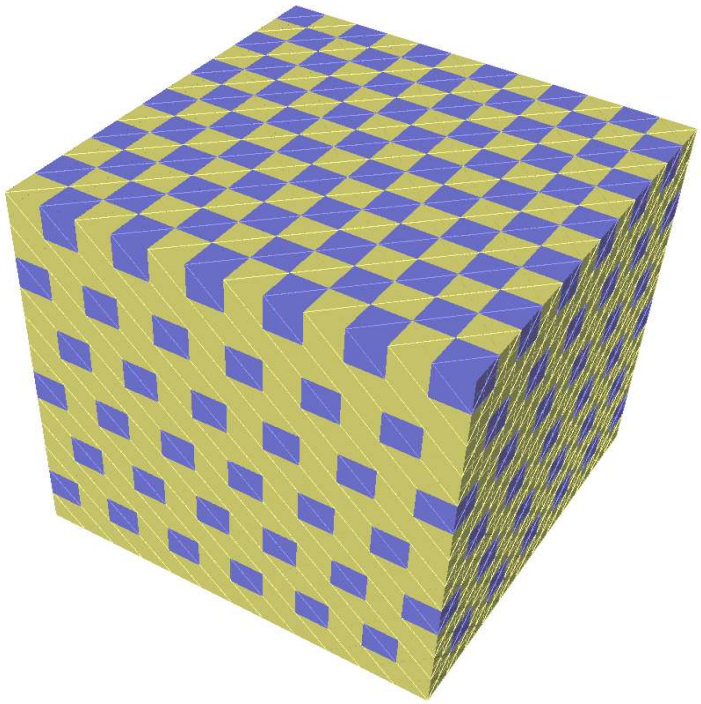

Figure 5.6
Alternating layers of a heterogeneous material distributed in a checkerboard pattern and a homogeneous, softer material.

Finally, we numerically check if primal vertex constraints are necessary at all. So far, all of our experiments were carried out without any primal vertices. In the theoretical analysis given in [84], it is shown that in some

Edge averages			Edge averages + 1st order moments		
κ	It.	Time	κ	It.	Time
2.14×10^5	$>$ **1 000**	$>$ 6 686s	5.19	**24**	629s

Table 5.3
Heterogeneous linear elasticity, see Figure 5.6 : Comparison of FETI-DP algorithm using edge averages vs. edge averages and first order moments carried out in parallel using 16 processors; we have $12 \times 12 \times 12 = 1\,728$ *cubic subdomains of* 5 184 *d.o.f. each, resulting in* 7 057 911 *total d.o.f. Stopping criterion: Relative residual reduction of* 10^{-10}.

Proc.	Subdomains/Proc.	It.	κ	Time
1	512	17	5.18	**1 828s**
2	256	17	5.18	**842s**
4	128	17	5.18	**428s**
8	64	17	5.18	**215s**
16	32	17	5.18	**122s**

Table 5.4
FETI-DP: Parallel scalability using edge averages and first order moments. 512 *subdomains with* 5 184 *d.o.f. each, yielding* 2 114 907 *global d.o.f. Stopping criterion: Relative residual reduction of* 10^{-7}.

special situations, primal vertices have to be introduced. We now construct such a model problem, following the theoretical considerations given in [84, Section 8.4]. We decompose the unit cube into $27 = 3 \times 3 \times 3$ cubic subdomains. The subdomains are made of two different materials, distributed such that subdomains of the same material type are only connected at the subdomain vertices; cf. Figure 5.7. From the results presented in Table 5.5, we conclude that there exist very hard cases of material distributions where we have to introduce primal vertices. This confirms the theoretical findings given in Klawonn and Widlund [84]. The problem considered in the present example is of course somewhat artificial. If primal vertices have to be introduced in real, industrial engineering applications in order to obtain a robust algorithm, still has to be numerically tested with such problems. First results for a non cubic geometry are given at the end of Section 5.5, where a hemisphere made of different materials is considered.

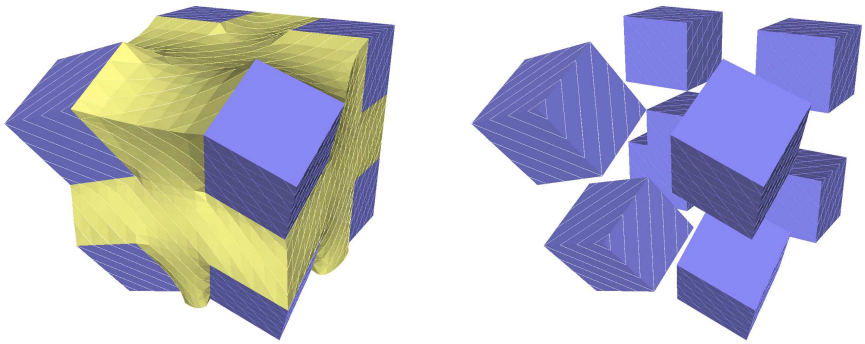

Figure 5.7
Unit cube decomposed into 9 stiff subdomains, sharing only vertices, and 18 soft subdomains. Left: Deformation showing stiff and soft subdomains. Right: Deformation, showing only stiff subdomains.

	Edge averages + primal vertices		Only edge averages	
E_1/E_2	It.	κ	It.	κ
1	26	**7.22**	26	7.56
10^3	23	**8.03**	49	89.58
10^6	24	**7.98**	113	8.38×10^4

Table 5.5
Comparison of FETI-DP using edge averages and vertices as primal constraints vs. a variant using only edge averages as primal constraints. Decomposition into $27 = 3 \times 3 \times 3$ subdomains, cf. Fig. 5.7, with 1 536 d.o.f. each, resulting in 31 944 global d.o.f. Stopping criterion: Relative residual reduction of 10^{-10}.

5.3 Acceptable Paths

In the general theory developed in [84], we do not have to make every face fully primal but it is sufficient to have an acceptable path; cf. the remark after Theorem 5.1.1 where the definition of an acceptable path is given. To illustrate numerically the effect of acceptable paths on the convergence behavior of our FETI-DP algorithms, we have carried out three different sets of experiments. In the first set of experiments, we considered the case of

two subdomains sharing a face, in the second two subdomains sharing an edge, and in the third two subdomains sharing only a vertex. In contrast to the model problems considered in the previous section, we now consider the effect of an acceptable path in the case where the shared face or edge is not fully primal, or the shared vertex is not primal; cf. Figure 5.8 and Table 5.6. From the numerical results in Table 5.6, we see that an acceptable path can help to control the condition number.

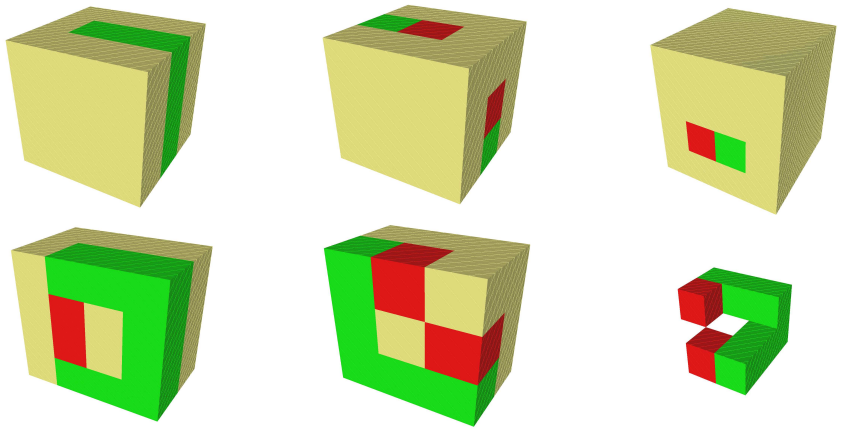

Figure 5.8
Cubes seen from the outside (upper images). Cubes cut open to see the two relevant subdomains and the path (lower images). Two stiff subdomains sharing a face, which are connected by a C-shaped path through 8 subdomains (left). Two stiff subdomains sharing an edge, which are connected by an L-shaped path through 5 subdomains (middle). Two stiff subdomains sharing a vertex, which are connected by a V-shaped path through 6 subdomains (right).

5.4 Curved Edges

In this section, we present numerical results for decompositions with curved edges. The theory in [84] is only for straight edges. We will see from the following numerical results that first order moments are still necessary in the case of large coefficient jumps.

In order to study the effect of slightly bent edges we start from the same configuration as in Table 5.2, i.e. with two stiff subdomains sharing an edge, surrounded by softer material with a Young's modulus $E_2 = 210$. We note

fully primal face		face not fully primal no path		face not fully primal path through 8 subdom.	
It.	κ	It.	κ	It.	κ
23	**5.33**	61	2.77×10^4	40	**23.04**
fully primal edge		edge not fully primal no path		edge not fully primal path through 5 subdom.	
It.	κ	It.	κ	It.	κ
24	**4.85**	48	5.80×10^3	31	**7.98**
primal vertex		vertex not primal no path		vertex not primal path through 6 subdom.	
It.	κ	It.	κ	It.	κ
26	**6.64**	36	4.84×10^2	27	**6.89**

Table 5.6

If the face, edge or vertex shared by the two stiff subdomains, see Fig. 5.8, is not made fully primal the condition number jumps dramatically, and the iteration count grows. If we have an acceptable path connecting the two stiff subdomains the condition number stays much smaller. The two special subdomains have $E_2 = 10^4 \times E_1$. We assume a stiffness of $E_3 = 10^6 \times E_1$ along the path. The iteration count is given for a relative residual reduction of 10^{-10}.

that in all of our experiments, the Poisson's ratio is $\nu = 0.29$. We then consider a slightly bent cross section, cf. Figure 5.9, according to an inner radius of curvature of $1/\sqrt{2}, 1$, and $\sqrt{2}$. We thus have a bent critical edge shared by the two stiff subdomains.

We start with a slightly bent configuration, corresponding to a radius of curvature of $1/\sqrt{2}$, as depicted in the leftmost picture of Figure 5.10. In Table 5.7, the effect is shown as we increase E_1, and the two subdomains become increasingly stiffer. We see that starting from a certain point we cannot control the condition number anymore by using only two first order moments on the edge. Nevertheless, if we choose the proper two moments, denoted as "good choice" in Table 5.7, the condition number is controlled much longer than in the case where we make the wrong choice, captioned "bad choice". If we use all three moments in the constraints for a bent edge then the condition number stays small independently of the ratio E_1/E_2.

Again, it is striking that the iteration counts in Table 5.7 do not reflect the growth of the corresponding condition numbers. We expect this again to be due to spectra which are nicely clustered except for some outliers which determine the high condition numbers. Additionally, in the cases where we

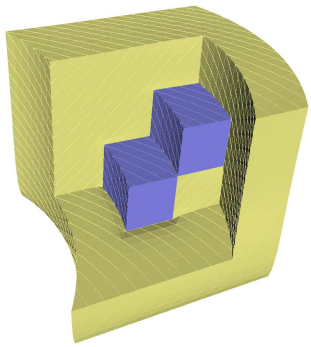

Figure 5.9
Deformed unit cube (radius of curvature = 1) with two deformed stiff subdomains sharing a curved edge surrounded by softer material.

	+2 (bad choice)			+2 (good choice)			+3		
E_1/E_2	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
10^0	35	13.03	1.0103	35	13.03	1.0103	35	**13.03**	1.0103
10^1	34	12.70	1.0098	34	12.70	1.0098	34	**12.70**	1.0098
10^2	35	12.60	1.0091	34	12.57	1.0093	34	**12.57**	1.0094
10^3	38	61.55	1.0083	35	12.53	1.0077	35	**12.53**	1.0078
10^4	43	5.71×10^2	1.0072	34	12.54	1.0082	34	**12.52**	1.0082
10^5	49	5.67×10^3	1.0064	35	43.67	1.0104	37	**12.51**	1.0069
10^6	54	5.67×10^4	1.0060	33	4.02×10^2	1.0156	39	**12.51**	1.0061

Table 5.7
Curved edge: Deformed unit cube decomposed into $3 \times 4 \times 4 = 48$ brick-shaped subdomains of 1 536 d.o.f. each. Radius of curvature $1/\sqrt{2}$, special edge is curved. 55 506 total d.o.f., relative tolerance 10^{-10}; 75 edges, edges use three edge averages. One special edge: +2, averages and, additionally, two first order moments; +3, averages and, additionally, three first order moments. Young's modulus $E_2 = 210$.

have large condition numbers, the residual oscillates strongly. This probably explains that in some cases with a high condition number the iteration count is smaller than for those with a small condition number. The same arguments apply for Table 5.8, see the remark below.

We then study the effect of different curvatures on the condition number and on the necessity to add all three moments to the constraints. In Table 5.8 we only consider three values for E_1/E_2, namely one, 10^3, and 10^6. We include the case $E_1/E_2 = 1$, i.e. the case of homogeneous material, to verify that the deformation of the unit cube only has a minor effect on the condition number and iteration count. This is the case whether we use two moments or three as long as we have a homogeneous material.

We also find that for $E_1/E_2 = 10^3$ we do not see any difference between using two and three constraints even if we bent the edge to a radius of curvature of $1/\sqrt{2}$ as depicted in the rightmost picture of Figure 5.10.

Only for $E_1/E_2 = 10^6$ do we see that two constraints are not sufficient anymore, and we have to introduce the third linearly independent moment as additional constraint in order to control the condition number and iteration count.

For the relation of condition numbers and iteration counts in Table 5.8, the same arguments as for Table 5.7 apply, see the discussion above.

From these numerical results, we conclude that discontinuities in the material stiffnesses of the order of 10^3 can still be treated using three edge averages and two first order moments. To obtain an unconditionally robust method, our experiments seem to indicate that, in the case of edges which are not straight, three edge averages and three first order moments are needed.

We have found that in decompositions coming from graph partitioners few edges have a multiplicity larger than three. This means that the number of edges is not very large where, potentially, first order moments have to be introduced, i.e. the situation in Figure 5.3. Thus we can afford to enforce moments whenever an edge has a multiplicity of larger than three.

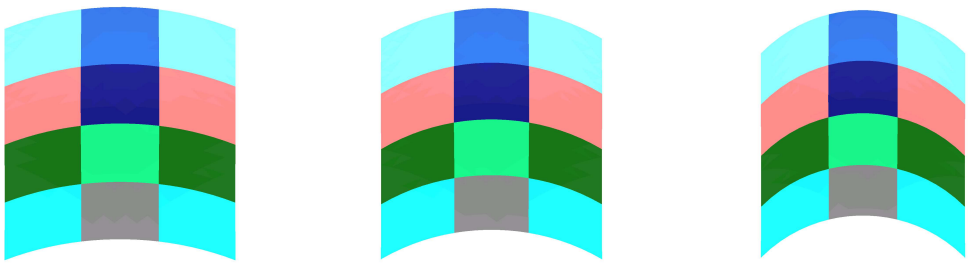

Figure 5.10
Curved configurations seen from bottom, radius of curvature $= \sqrt{2}, 1, 1/\sqrt{2}$.

	Radius of		+2			+3	
E_1/E_2	curvature	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
	straight	28	9.19	1.0113			
	$\sqrt{2}$	31	9.99	1.0106	31	**9.99**	1.0106
10^0	1	32	10.66	1.0105	32	**10.66**	1.0105
	$1/\sqrt{2}$	35	13.03	1.0103	35	**13.03**	1.0103
	straight	30	9.11	1.0084			
	$\sqrt{2}$	32	9.80	1.0075	31	**9.80**	1.0082
10^3	1	33	10.45	1.0074	32	**10.45**	1.0080
	$1/\sqrt{2}$	35	12.53	1.0077	35	**12.53**	1.0078
	straight	30	9.10	1.0080			
	$\sqrt{2}$	28	1.01×10^2	1.0153	35	**9.79**	1.0065
10^6	1	30	2.00×10^2	1.0150	35	**10.45**	1.0063
	$1/\sqrt{2}$	33	4.02×10^2	1.0156	39	**12.51**	1.0061

Table 5.8

Curved edge: Deformed unit cube decomposed into $3 \times 4 \times 4 = 48$ *brick-shaped subdomains of* 1 536 *d.o.f. each. Radius of curvature* $1/\sqrt{2}$*, special edge is curved.* 55 506 *total d.o.f., relative tolerance* 10^{-10}*; 75 edges, edges use three edge averages. One special edge: +2, averages and, additionally, two first order moments; +3, averages and, additionally, three first order moments. Young's modulus* $E_2 = 210$*.*

5.5 Material Heterogeneities Not Aligned with the Interface

In the theoretical estimates presented in [84], it is assumed, as it is standard in theoretical analyses of this type, that the coefficient jumps of the Young's moduli are aligned with the interface, i.e. discontinuities can only occur across the subdomain boundaries. In practice, satisfying such an assumption can lead to a decomposition with very bad aspect ratios which usually spoil the convergence rate. In this section, we numerically analyze the effect of material heterogeneities which are not aligned with the interface. We first apply our algorithm with edge averages, but without first order moments

and primal vertices, to four different model problems. We always consider a unit cube which is fixed at one face. In our experiments we always compare the case of a homogeneous material to different distributions of materials with different stiffnesses. We assume that the softer material has a Young's modulus of $E_2 = 210$ and a Poisson's ratio of $\nu = 0.29$. The Young's modulus of the stiffer material is denoted by E_1 and the ratio of the two is $E_1/E_2 = 10^6$. We use four node tetrahedral elements, a decomposition into $2\times2\times2 = 8$ cubic subdomains with 89 373 d.o.f. each, resulting in 680 943 global d.o.f. As stopping criterion, we use the relative residual reduction of 10^{-10}. In the first three experiments, the parts of Ω which consist of the stiffer material do not intersect any interior edges; cf. Figures 5.11, 5.12, 5.13, and 5.14. We note that these examples are constructed such that the material cannot be treated by the standard scaling as described in Section 2.3 since the material discontinuity is not across the interface. From the results given in Tables 5.9, 5.10, and 5.11, we see that such a jump in the Young's modulus seems not to affect severely the condition and iteration number.

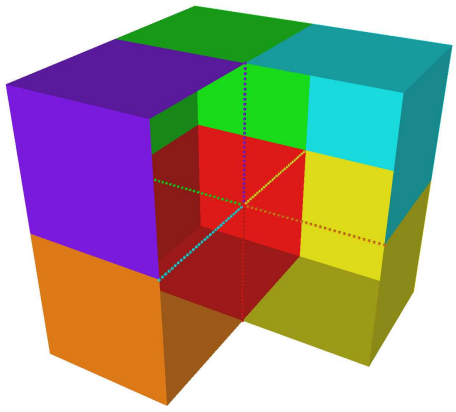

Figure 5.11
For $2 \times 2 \times 2 = 8$ subdomains, the 6 edges in the interior of Ω are shown (two subdomains in front of the cube are not displayed).

Next, we consider an example where a stiff cube is located at the center of a larger cube made out of a softer material; cf. Figure 5.15. The essential difference in comparison to the previous examples is that the stiffer subcube now intersects the interior edges; see also Figure 5.11. From the results given in the mid-column of Table 5.12, we see that this severely affects the iteration and condition number. As a remedy, we introduce a weighted edge average

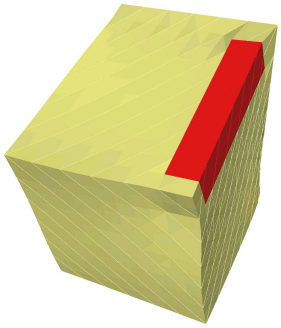

Figure 5.12
Material heterogeneities not aligned with the interface. Soft cube (Young's modulus $E_2 = 210$) with a stiff beam (Young's modulus $E_1 = 10^6 \times E_2$), with square cross section at the upper right edge.

Homogeneous		Heterogeneous	
It.	κ	It.	κ
31	18.62	42	32.59

Table 5.9
Iteration counts and condition number estimates for problem given in Figure 5.12. Homogeneous: $E_1 = E_2 = 210$. Heterogeneous: $E_2 = 210, E_1 = 10^6 \times E_2$.

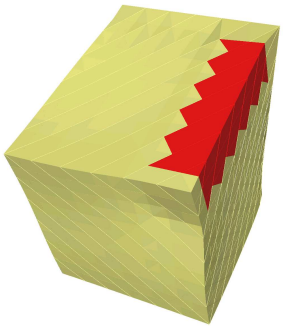

Figure 5.13
Material heterogeneities not aligned with the interface. Soft cube (Young's modulus $E_2 = 210$) with a stiff beam (Young's modulus $E_1 = 10^6 \times E_2$), with square cross section and a jagged interface at the upper right edge.

Homogeneous		Heterogeneous	
It.	κ	It.	κ
31	18.62	43	34.16

Table 5.10
Iteration counts and condition number estimates for problem given in Figure 5.13. Homogeneous: $E_1 = E_2 = 210$. Heterogeneous: $E_2 = 210, E_1 = 10^6 \times E_2$.

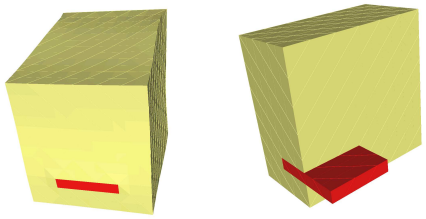

Figure 5.14
Material heterogeneities not aligned with the interface. Soft cube (Young's modulus $E_2 = 210$) with a stiff rectangular plate (Young's modulus $E_1 = 10^6 \times E_2$) which reaches into the cube and is only shared by the two lower subdomains at the front.

Homogeneous		Heterogeneous	
It.	κ	It.	κ
31	18.62	40	22.99

Table 5.11
Iteration counts and condition number estimates for problem given in Figure 5.14. Homogeneous: $E_1 = E_2 = 210$. Heterogeneous: $E_2 = 210, E_1 = 10^6 \times E_2$.

of the form

$$\frac{\sum_{x_i \in \mathcal{E}^h} \rho(x_i)\, u_j(x_i)}{\sum_{x_i \in \mathcal{E}^h} \rho(x_i)}, \quad j = 1, 2, 3, \tag{5.2}$$

with weights $\rho(x_i)$ defined pointwise by the maximum material stiffness at that point and $u = [u_1^T, u_2^T, u_3^T]^T$. We note that this weighted edge average is reduced to the standard edge average in the case of material jumps aligned with the interface. The results given in the column on the right hand side of Table 5.12 indicate that this weighted edge average could be helpful in cases where the material discontinuities do not align with the interface. This should be further analyzed for more difficult problems coming from real-world, engineering examples.

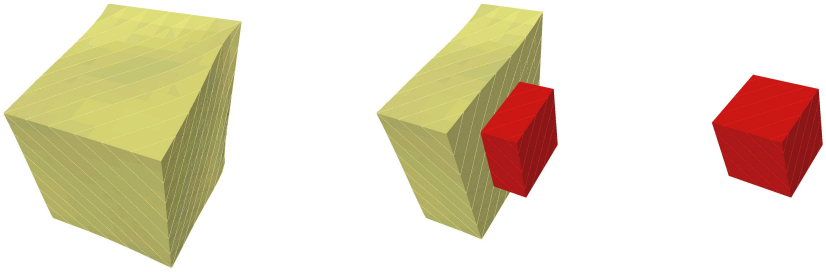

Figure 5.15
Material heterogeneities not aligned with the interface. Soft material (Young's modulus $E_2 = 210$) surrounding a stiffer cube (Young's modulus $E_1 = 10^6 \times E_2$), centered at the origin of the cube. The stiffer, interior cube intersects all interior edges.

Edge average (hom.)		Edge average (het.)		Weighted average (het.)	
It.	κ	It.	κ	It.	κ
31	18.62	179	4.11×10^6	37	16.57

Table 5.12
Comparison of standard and weighted edge averages for the problem given in Figure 5.15. Decomposition into $2 \times 2 \times 2 = 8$ cubic subdomains with 89 373 d.o.f. each, yielding 680 943 global d.o.f. Stopping criterion: Relative residual reduction of 10^{-10}. Primal constraints: Edge averages.

Next, we present a small example with a heterogeneous material and a non cubic geometry constructed using Netgen [116], see Figure 5.16. It

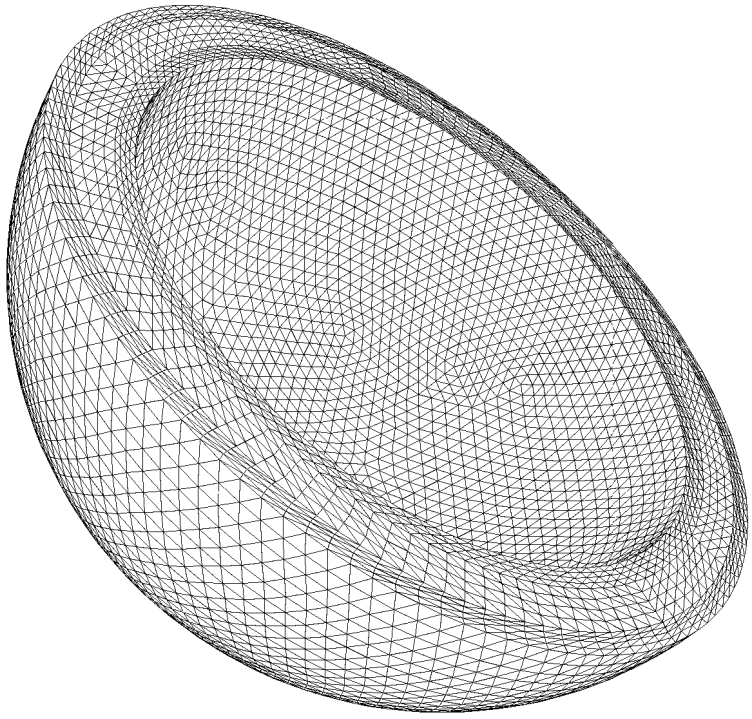

Figure 5.16: *Hemisphere with three different material layers.*

is a hemisphere consisting of three material layers. The outer and inner layer have a Young's modulus of $E_2 = 210$ whereas the interior layer has a Young's modulus of $E_1 = 10^6 \times E_2$. We impose homogeneous Dirichlet boundary conditions on the interior layer where it intersects the boundary of the hemisphere; a volume force is applied. The model has 209 768 tetrahedral elements and 116 055 d.o.f. The grid is partitioned into 20 subdomains using ParMetis [70] not taking into account the material distribution. The jumps in the Young's modulus are therefore not aligned with the subdomain boundaries. The coarse problem has only 30 d.o.f. We have a condition number of 75.62 and an iteration count of 53 for a relative residual reduction of 10^{-7}. The solution time is $27s$ on 5 processors of our Opteron computing cluster. This result indicates that our strategies may be useful for real world heterogeneous models; this of course still remains to be tested for such problems.

5.6 Conclusions

We have presented a FETI-DP algorithm for elasticity problems with large jumps in the material stiffness. This method was introduced and theoretically analyzed in [84], and the results shown in Section 5.2 for straight edges confirm the theoretical findings in [84]. We also considered curved edges in Section 5.4, and the results show that for moderate jumps in the Young's modulus and slightly bent edges it is sufficient to use two first order moments. For arbitrarily large jumps, we advocate the use of all three first order moments. Another important situation, which usually cannot be covered by theoretical investigations, is the case when the material discontinuities are not aligned with the interface. Our numerical results indicate that such a material distribution seems not to affect the condition number and the iteration count if the discontinuity does not appear on an interior edge. If the discontinuity appears on an interior edge, in the experiment considered here, a weighted edge average was a remedy. Since this weighted average can be implemented without additional cost, we suggest it as the default setting. Further, extensive numerical tests for real-world, industrial engineering problems should be pursued. First results for a hemisphere consisting of different material layers are also provided.

Chapter 6

Almost Incompressible Elasticity Problems

In finite element approximations for problems in solid mechanics volumetric locking can appear. This can especially be the case when incompressible materials are discretized with standard low order finite element discretizations, which do not ensure uniform convergence in the incompressible limit. Methods where the strain or stress field is enriched by the addition of carefully chosen basis functions, see, e.g., [124], are popular and have proven to be highly effective. In this section, we are interested in pure displacement based formulations, which are obtained by local static condensation of a mixed problem satisfying a uniform inf-sup condition.

This section is based on joint work with Wohlmuth and Klawonn and has been published in a proceedings paper [79]. Here, we work with conforming bilinear approximations for the displacement and a discontinuous pressure space for almost incompressible elasticity. Unfortunately, the standard Q1-P0 pairing does not satisfy a uniform inf-sup condition. To obtain a stable scheme, we have to extract from the pressure space the so-called checkerboard modes. For some earlier references on the construction of uniformly bounded domain decomposition and multigrid methods in the incompressible limit, see the work for Goldfeld [54, 55] for Neumann-Neumann methods and [144] and [117] for multigrid solvers. Let us note that there are also recent results on FETI-DP and BDDC domain decomposition methods for mixed finite element discretizations of Stokes' equations, see [90] and [89], almost incompressible elasticity, see Dohrmann [30], and on incompressible elasticity, see Dohrmann and Lehoucq [32]. We propose a dual-primal iterative substructuring method for almost incompressible elasticity working with a pure displacement formulation after having eliminated all pressure variables. Numerical results illustrate the performance and the scalability of

our method in the incompressible limit. There is a close relationship to the work done by Dohrmann for BDDC [30].

6.1 Discretization

We recall the weak formulation of linear elasticity

$$2\mu \int_\Omega \varepsilon(u) : \varepsilon(v)\, dx + \lambda \int_\Omega \mathrm{div}(u)\mathrm{div}(v)\, dx = \langle \boldsymbol{F}, v\rangle \ \ \forall v \in \mathbf{H_0^1}(\Omega, \partial\Omega_D), \tag{6.1}$$

see Section 3.1. The parameters μ and λ are the positive Lamé constants.

Until now, we have only considered compressible linear elasticity but will now turn to the incompressible limit. It corresponds to $\lambda \to \infty$. The Lamé parameters are related to the pair (E, ν), where E is Young's modulus and ν is Poisson's ratio by

$$E = \frac{\mu(2\mu + 3\lambda)}{\mu + \lambda}, \quad \nu = \frac{\lambda}{2(\mu + \lambda)}$$

or

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

cf. Section 3.1.

The finite element discretization is based on the conforming space $\mathbf{V}_h$ of continuous piecewise bilinear approximations on quadrilaterals. The quasi-uniform mesh is denoted by $\mathcal{T}_h$, and we assume that it has a macro-element structure, i.e. $\mathcal{T}_h$ is obtained by uniform refinement from a coarser mesh $\mathcal{T}_h^m$. To start with, we consider the abstract pair $(\mathbf{V}_h, M_h)$

$$\begin{array}{lcll} 2\mu(\varepsilon(u_h), \varepsilon(v_h))_0 & + & (\mathrm{div}\, v_h, p_h)_0 & = \langle F, v_h\rangle \quad \forall v_h \in \mathbf{V}_h\ , \\ (\mathrm{div}\, u_h, q_h)_0 & - & \frac{1}{\lambda}(p_h, q_h)_0 & = 0 \qquad\quad \forall q_h \in M_h\, . \end{array}$$

In terms of static condensation, we can eliminate the pressure and obtain a displacement based formulation

$$\int_\Omega 2\mu\varepsilon(u) : \varepsilon(v)\, dx + \int_\Omega \lambda\, \Pi_{M_h}\mathrm{div}\, u\ \Pi_{M_h}\mathrm{div}\, v\, dx = \langle F, v\rangle \ \ \forall v \in \mathbf{V}_h, \tag{6.2}$$

where Π_{M_h} denotes the L^2-projection onto M_h. It is well known that the choice $M_h = M_h^u$,

$$M_h^u = \{q \in L_0^2(\Omega) : q|_K \in P_0(K),\ K \in \mathcal{T}_h\},$$

does not yield a uniform inf-sup condition, and checkerboard modes in the pressure might be observed; see, e.g., [53]. Thus it is necessary to make M_h a proper subset of M_h^u. There exist different possibilities to overcome this difficulty. One option is to work with macro-elements and to extract from M_h^u the checkerboard mode on each macro-element, as in [53]. The restrictions of functions in M_h^u to a macro-element are spanned by the four functions depicted in Figure 6.1.

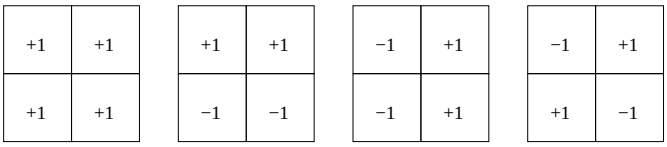

Figure 6.1
Restrictions of the basis functions of M_h^u to a macro-element with ± 1 indicating the sign inside the elements.

The functions having the signs indicated in Figure 6.1 are the local checkerboard modes p^c. To obtain a stable pairing, we have to work with $M_h = M_h^s$,

$$M_h^s = \{q \in M_h^u : (q, p^c)_{0;K} = 0,\ K \in \mathcal{T}_h^m\}.$$

From now on, we call the choice $M_h = M_h^u$ the unstable or the not stabilized Q1-P0 formulation and the choice $M_h = M_h^s$ the stabilized Q1-P0 formulation. The analysis and the implementation will be based on the reduced problem (6.2). We note that in both cases the L^2-projection Π_{M_h} can be carried out locally.

6.2 Coarse Problem

We have to decide how to choose the primal displacement variables. The simplest choice is to choose them as certain selected vertices of the subdomains, i.e. Algorithm A. Unfortunately, Algorithm A does not yield uniform bounds in the incompressible limit. To obtain better convergence properties, we have to introduce additional constraints. These constraints are averages over the edges, which are enforced to have the same values across the interface. This variant has been introduced in [86] for scalar problems and is denoted by Algorithm B.

For our FETI-DP Algorithm B, we expect the following condition number estimate, Theorem 6.2.1.

Theorem 6.2.1 *The condition number for the choice $M_h = M_h^s$ satisfies*

$$\kappa(M^{-1}F) \leq C\,(1 + \log(H/h))^2.$$

Here, $C > 0$ is independent of h, H, and the values of the Poisson ratio ν.

6.3 Numerical Results

We apply Algorithms A and B to (6.2), where $\Omega = (0,1)^2$, and the Young's modulus is defined as $E = 1$. We will present results for different Poisson ratios ν. Algorithm A uses all subdomain vertices as primal constraints and Algorithm B, additionally, introduces edge averages into the primal constraints. For the experiments in Table 6.1, we use a structured grid with 240×240 macro elements ($= 480 \times 480$ elements). In small portions of the boundary in all four corners of the unit square homogeneous Dirichlet boundary conditions were applied, see Figure 6.2, and the domain was subjected to a volume force directed towards $(1,1)^T$. The domain was decomposed into 64 square subdomains with 7 442 d.o.f. each; this results in an overall problem with 462 722 d.o.f. The stopping criterion is a relative residual reduction of 10^{-10}. The experiments were carried out on two Opteron 248 (2.2 GHz) 64-bit processors. The differences in computing time between the unstable and the stabilized Q1-P0 element, e.g. for $\nu = 0.4$, are due to the different sparsity patterns of the stiffness matrices. The stabilized Q1-P0 element leads to up to 50 percent more nonzero entries in the corresponding stiffness matrix.

For the experiments in Table 6.2, the unit square is decomposed into 4 to 1 024 subdomains with 1 250 d.o.f. each. Homogeneous Dirichlet boundary conditions are applied on the bottom and the left side. Again, a volume force directed towards $(1,1)^T$ is applied. The calculations were carried out on a single AMD Opteron 144 (1.8 GHz) 64-bit processor. We used as a stopping criterion the relative residual reduction of 10^{-14}. Here, as usual, we mean the pseudo-residual which is calculated recursively in the conjugate gradient method.

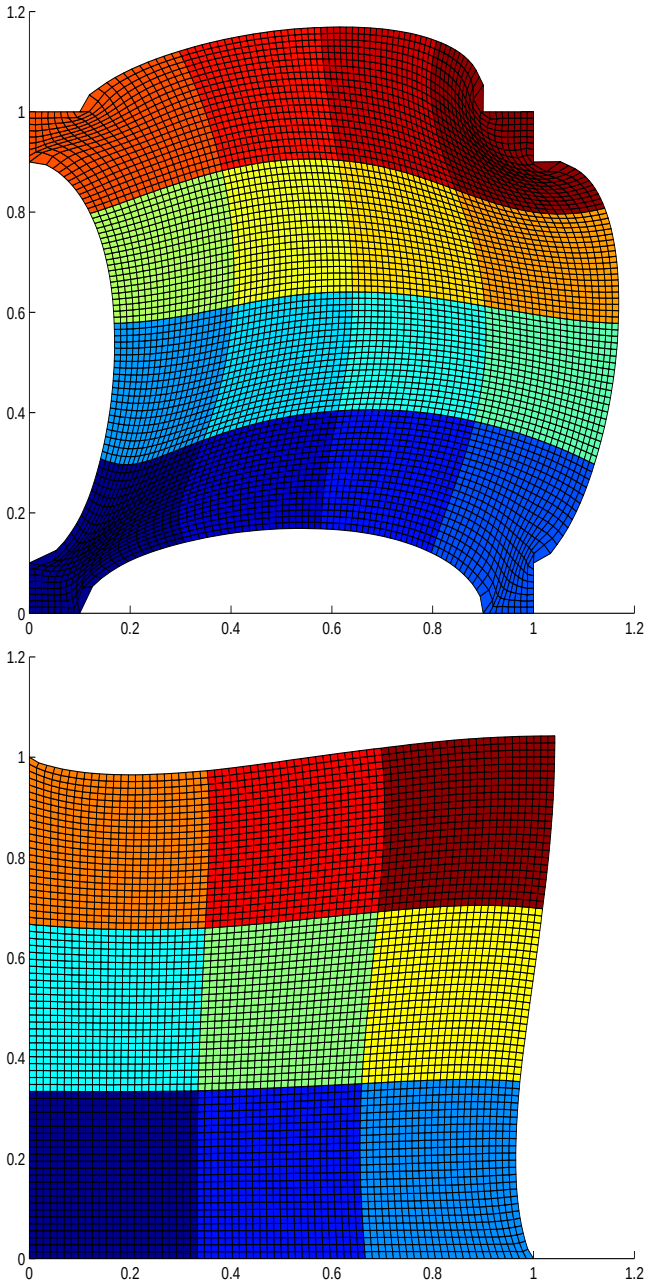

Figure 6.2

Deformed configuration for the experiments in Table 6.1 (upper) and for the experiments in Table 6.2 (lower). In the calculations finer grids are used than the ones depicted here.

ν	It.	$\lambda_{\max}$	$\lambda_{\min}$	Time	It.	$\lambda_{\max}$	$\lambda_{\min}$	Time
Alg. B		(**stabilized**)				(**not stabilized**)		
0.4	23	**6.98**	1.0075	55s	23	**6.98**	1.0075	47s
0.49	23	**6.81**	1.0079	55s	23	**6.86**	1.0086	47s
0.499	24	**6.79**	1.0078	56s	23	**6.79**	1.0090	47s
0.4999	24	**6.79**	1.0078	56s	29	**6.48**	1.0087	53s
0.49999	24	**6.79**	1.0080	56s	55	**39.98**	1.0088	80s
0.499999	25	**6.79**	1.0076	57s	97	**366**	1.0086	124s
0.4999999	25	**6.79**	1.0078	57s	131	**3 632**	1.0096	159s
Alg. A		(**stabilized**)				(**not stabilized**)		
0.4	53	**42.52**	1.012	82s	53	**42.52**	1.012	81s
0.49	103	**316**	1.017	139s	67	**85.93**	1.015	78s
0.499	192	**3 037**	1.018	241s	137	**723**	1.017	143s
0.4999	270	$\mathbf{3.02} \times 10^4$	1.020	332s	220	**7 069**	1.020	221s
0.49999	368	$\mathbf{3.02} \times 10^5$	1.020	445s	315	$\mathbf{7.05} \times 10^4$	1.021	310s
0.499999	465	$\mathbf{3.02} \times 10^6$	1.022	558s	>500	$\mathbf{7.05} \times 10^5$	1.037	>486s
0.4999999	>500	$\mathbf{3.02} \times 10^7$	1.032	>599s	>500	$\mathbf{7.05} \times 10^6$	1.159	>484s

Table 6.1: *Algorithms B and A, 462 722 d.o.f. and 64 subdomains.*

Algorithm B			$\nu = \mathbf{0.4999999}$			$\nu = \mathbf{0.4}$		
N	Mesh	d.o.f.	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
4	48×48	4 802	17	2.51	1.0011	13	2.19	1.0015
9	72×72	10 658	21	3.38	1.0020	19	3.47	1.0024
16	96×96	18 818	24	4.03	1.0023	22	4.13	1.0025
36	144×144	42 050	26	4.53	1.0024	24	4.64	1.0025
64	192×192	74 498	27	4.69	1.0024	25	4.80	1.0026
100	240×240	116 162	29	4.75	1.0022	26	4.86	1.0025
144	288×288	167 042	29	4.78	1.0023	27	4.88	1.0026
256	384×384	296 450	30	4.79	1.0022	30	4.91	1.0024
576	576×576	665 858	32	4.80	1.0021	32	4.77	1.0024
1 024	768×768	1 182 722	32	4.80	1.0021	33	4.81	1.0024

Table 6.2: *Numerical scalability of Algorithm B, Q1-P0 (stabilized).*

Chapter 7

Higher Order Methods

7.1 Spectral Elements

The following section reports on results of joint work with Pavarino and Klawonn [74]. In higher-order finite element methods like spectral elements or the *hp*-version finite elements the accuracy of the discrete solution is improved by increasing the polynomial degree of the basis functions as well as the number of elements. The *hp* finite elements are usually based on hierarchical non-nodal basis functions. They have been studied mostly in the structural mechanics community; see, e.g., Szabó and Babuška [132] and Schwab [119]. On the other hand, spectral elements are based on tensorial nodal bases associated with Gauss-Lobatto-Legendre (GLL) quadrature nodes. They have been studied mostly in the fluid dynamics community; see, e.g., Canuto, Hussaini, Quarteroni, and Zang [23], Bernardi and Maday [12], Funaro [49], Karniadakis and Sherwin [68], Deville, Fischer, and Mund [28].

Higher-order methods generate linear systems which are much more ill-conditioned than the ones obtained from standard low-order finite elements. The condition number of these higher-order discrete systems is still proportional to the square of the element size but can be proportional to the cube or fourth power of the polynomial degree of the basis functions. Therefore, it is important to construct efficient preconditioners for these methods. Particularly open to research are preconditioners for nontensorial spectral and *hp*-finite elements, usually on triangular and tetrahedral elements, see Ainsworth [3], Bica [15], Pavarino and Warburton [105], Sherwin and Casarin [122], Giraldo and Warburton [52], Pasquetti et al. [103], and Schöberl et al. [118]. In this section, we will focus on tensorial GLL spectral elements only.

In this section, we consider the FETI-DP method [40, 84, 76], as well as its primal counterparts, the algorithms known as the Balancing Domain

Decomposition methods by Constraints (BDDC) method [29, 26, 97, 98, 92], cf. Chapter 2.

In both methods the condition number only depends weakly on the polynomial degree. We will also consider the inexact versions of the FETI-DP methods that were introduced in Chapter 4. Inexact BDDC methods have been considered in [139] and recently in [91] and [31] for low order finite elements.

7.1.1 Discretization

Let T_{ref} be the reference square $(-1,1)^2$, and let $Q_p(T_{\text{ref}})$ be the set of polynomials on T_{ref} of degree $p \geq 1$ in each variable. We assume that the domain Ω can be decomposed into N_e nonoverlapping finite elements T_k of characteristic diameter h,

$$\overline{\Omega} = \bigcup_{k=1}^{N_e} \overline{T}_k,$$

each of which is an affine image of the reference square or cube, $T_k = \phi_k(T_{\text{ref}})$, where ϕ_k is an affine mapping (more general maps could be considered as well). Then, we will group these elements into N nonoverlapping subdomains Ω_i of characteristic diameter H, forming themselves a coarse finite element partition of Ω,

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i, \qquad \overline{\Omega}_i = \bigcup_{k=1}^{N_i} \overline{T}_k.$$

Hence, the fine element partition $\{T_k\}_{k=1}^{N_e}$ can be considered a refinement of the coarse subdomain partition $\{\Omega_i\}_{i=1}^{N}$, with matching finite element nodes on the boundaries of neighboring subdomains.

We consider linear, selfadjoint, elliptic problems on Ω, with zero Dirichlet boundary conditions on a part $\partial\Omega_D$ of the boundary $\partial\Omega$:

Find $u \in V = \{v \in H^1(\Omega) : \ v = 0 \ \text{ on } \ \partial\Omega_D\}$ such that

$$a(u,v) = \int_\Omega \rho(x) \nabla u \cdot \nabla v \, dx \ = \ \int_\Omega f v \, dx \qquad \forall \ v \in \ V. \tag{7.1}$$

Here, $\rho(x) > 0$ can be discontinuous with very different values for different subdomains but we assume this coefficient to vary only moderately within each subdomain Ω_i. In fact, without decreasing the generality of our results, we will only consider the piecewise constant case of $\rho(x) = \rho_i$, for $x \in \Omega_i$.

Conforming spectral elements discretizations consists of continuous, piecewise polynomials of degree p in each element:

$$V^p = \{v \in V : v|_{T_i} \circ \phi_i \in Q_p(T_{\text{ref}}), \ i = 1, \cdots, N_e\}.$$

A convenient tensor product basis for $\mathbf{V}^p$ is constructed using Gauss-Lobatto-Legendre (GLL) quadrature points; other bases could be considered, such as those based on integrated Legendre polynomials common in the p-version finite element literature; see Szabó and Babuška [132]. Let $\{\xi_i\}_{i=0}^p$ denote the set of GLL points on $[-1,1]$, and let σ_i denote the quadrature weight associated with ξ_i. Let $l_i(\cdot)$ be the Lagrange interpolating polynomial which vanishes at all the GLL nodes except ξ_i, where it equals one. The basis functions, e.g. on the reference square, are then defined by a tensor product as

$$l_i(x_1)l_j(x_2), \qquad 0 \le i,j \le p.$$

This basis is nodal since every element of $Q_p(T_{\text{ref}})$ can be written as

$$u(x_1,x_2) = \sum_{i=0}^{p}\sum_{j=0}^{p} u(\xi_i,\xi_j)l_i(x_1)l_j(x_2).$$

Each integral of the continuous model (7.1) is replaced by GLL quadrature. On T_{ref},

$$(u,v)_{p,T_{\text{ref}}} = \sum_{i=0}^{p}\sum_{j=0}^{p} u(\xi_i,\xi_j)v(\xi_i,\xi_j)\sigma_i\sigma_j,$$

and on all of Ω,

$$(u,v)_{p,\Omega} = \sum_{k=1}^{N_e}\sum_{i,j=0}^{p}(u\circ\phi_k)(\xi_i,\xi_j)(v\circ\phi_k)(\xi_i,\xi_j)|J_k|\sigma_i\sigma_j,$$

where $|J_k|$ is the determinant of the Jacobian of ϕ_k. This inner product is uniformly equivalent to the standard L_2-inner product on $Q_p(T_{\text{ref}})$:

$$\|u\|^2_{L_2(T_{\text{ref}})} \le (u,u)_{p,T_{\text{ref}}} \le C\|u\|^2_{L_2(T_{\text{ref}})} \qquad \forall u \in Q_p(T_{\text{ref}}), \tag{7.2}$$

see Bernardi and Maday [12]. These bounds imply an analogous uniform equivalence between the $H^1(\Omega)$-seminorm and the discrete seminorm $(\nabla u,\nabla u)_{n,\Omega}$ based on GLL quadrature. Applying these quadrature rules, we obtain the discrete bilinear form

$$a_p(u,v) = \sum_{k=1}^{N_e}(\rho_k\nabla u,\nabla u)_{p,T_k}$$

and the discrete elliptic problem:
Find $u \in V^p$ such that

$$a_p(u,v) = (f,v)_{p,\Omega} \qquad \forall v \in V^p. \tag{7.3}$$

Having chosen a basis for V^p, the discrete problem (7.3) is then turned into a linear system of algebraic equations

$$K_g u_g = f_g, \tag{7.4}$$

where K_g is the globally assembled, symmetric, positive definite stiffness matrix. We have denoted with the same symbols u_g and f_g the vectors representing the corresponding spectral element functions in the given basis.

7.1.2 Convergence Estimate

As shown in Toselli and Widlund [136] for the two main families of overlapping Schwarz methods (Chapter 7.3) and iterative substructuring methods of wirebasket and Neumann-Neumann type (Chapter 7.4), the main domain decomposition results obtained for finite element discretizations of scalar elliptic problems can be transferred to the spectral element case using some results by Canuto [22], Bernardi and Maday [11], and Casarin [24]. The same tools can be used here, thus extending the main result for BDDC (see [97], Mandel, [98], [92]) and FETI-DP, see [39], [86], [136, Chapter 6]) from the finite element case to the spectral element case. Direct proofs have also been given for some Neumann-Neumann and wirebasket methods, see, e.g., [104, 106] and the references therein.

In this section, we employ the FETI-DP and BDDC method as described in Chapter 2 and the irFETI-DP method as described in Chapter 4.

Theorem 7.1.1 *Under the assumptions in this section, the BDDC and FETI-DP preconditioned operators have the same spectrum, except for the eigenvalue one. The minimum eigenvalue is bounded by one and maximum eigenvalue bounded by*

$$C\left(1+\log\left(p\,\frac{H}{h}\right)\right)^2,$$

with $C>0$ independent of p, h, H and the values of the coefficients ρ_i of the elliptic operator.

We remark that this result depends on the tensorial structure of classical spectral elements based on GLL nodal bases. For nontensorial spectral and *hp* elements on triangles or tetrahedra, the construction and analysis of efficient preconditioners remains in most cases an open problem. Nevertheless, some results are available in Ainsworth [3], Bica [15], and Schöberl et al. [118], and numerical studies can be found in Pavarino and Warburton [105], Sherwin and Casarin [122], and Pasquetti et al. [103].

7.1.3 Numerical Results for FETI-DP and BDDC

We first investigate the growth of the condition number for an increasing number of subdomains. Note that we always have the lower bound of one for the smallest eigenvalue. We expect to see the largest eigenvalue, and thus also the condition number, approach a constant value, independent of coefficient jumps but dependent on the polynomial degree. In Tables 7.1 and 7.2 and Figure 7.1, we see the expected behavior for different polynomial degrees and fixed $H/h = 1$. In Tables 7.3, 7.4 and Figure 7.3, we see that this is also the case for $H/h = 2$.

From these results, we use a number of $N \geq 256$ subdomains in our experiments to study the asymptotic behavior of the condition number. In Table 7.5 and Figure 7.3, we choose a sufficient number of subdomains and increase the polynomial degree from 2 to 32. We see that the condition number grows only slowly and, from Figure 7.3 (lower), we see that we indeed have the expected $C(1 + \log(p))^2$-bound.

In Table 7.6 and Figure 7.4, we then keep the polynomial degree fixed to $p = 4$ and $p = 8$ and increase H/h from 1 to 32. Again, the condition number behaves as expected and the $C(1 + \log(H/h))^2$ bound is confirmed from Figure 7.4 (lower). Thus, we have confirmed a bound on the condition number of $C(1 + \log(pH/h))^2$.

In Table 7.6, we have also shown the CPU timings and iteration counts of irFETI-DP, see Chapter 4, additionally to FETI-DP. From the table we see that also for spectral elements irFETI-DP compares very well with standard FETI-DP. Here, we again have used BoomerAMG [66] for the coarse problem, cf. Section 4.9. Only for completeness, do we report on the parallel scalability for 2 to 16 processors in Table 7.7 for FETI-DP and irFETI-DP. Both methods show basically the same performance and same scalability also for spectral elements.

		Unpreconditioned Schur Complement			BDDC			FETI-DP		
p	N	It.	λmax	λmin	It.	λmax	λmin	It.	λmax	λmin
2	4	**3**	4.89	1.3333	**2**	1.05	1	**2**	1.05	1
	16	**15**	5.47	0.3905	**6**	1.45	1.0006	**6**	1.45	1.0026
	64	**33**	5.62	0.1015	**8**	1.61	1.0010	**8**	1.61	1.0014
	256	**66**	5.66	0.0256	**8**	1.64	1.0010	**8**	1.64	1.0028
	576	**97**	5.67	0.0114	**8**	1.65	1.0012	**8**	1.66	1.0032
3	4	**7**	5.44	0.7823	**3**	1.21	1	**3**	1.21	1
	16	**21**	5.73	0.2332	**8**	2.10	1.0010	**8**	2.10	1.0007
	64	**26**	5.81	0.0608	**11**	2.32	1.0005	**11**	2.32	1.0006
	256	**86**	5.83	0.0154	**11**	2.37	1.0005	**11**	2.37	1.0006
	576	**129**	5.83	0.0068	**11**	2.38	1.0004	**11**	2.38	1.0006
4	4	**10**	5.60	0.5440	**3**	1.37	1	**3**	1.37	1
	16	**25**	5.78	0.1655	**9**	2.65	1.0011	**9**	2.65	1.0018
	64	**51**	5.82	0.0433	**13**	2.95	1.0013	**12**	2.95	1.0022
	256	**100**	5.84	0.0110	**13**	3.02	1.0011	**13**	3.01	1.0020
	576	**146**	5.84	0.0049	**13**	3.02	1.0010	**13**	3.03	1.0020
8	4	**8**	5.80	0.2381	**4**	1.89	1	**4**	1.89	1
	16	**38**	5.86	0.0761	**12**	4.38	1.0008	**12**	4.38	1.0007
	64	**75**	5.88	0.0202	**17**	4.88	1.0015	**16**	4.86	1.0013
	256	**143**	5.88	0.0051	**17**	5.00	1.0016	**17**	5.00	1.0014
	576	**208**	5.89	0.0023	**17**	5.01	1.0017	**17**	5.01	1.0015
16	4	**29**	5.95	0.1099	**5**	2.57	1	**5**	2.57	1
	16	**59**	5.97	0.0364	**14**	6.65	1.0009	**14**	6.65	1.0009
	64	**110**	5.97	0.0097	**21**	7.42	1.0012	**21**	7.42	1.0013
	256	**206**	5.97	0.0025	**21**	7.59	1.0012	**21**	7.58	1.0017
	576	**307**	5.97	0.0011	**21**	7.61	1.0012	**21**	7.62	1.0016
32	4	**43**	6.09	0.0525	**6**	3.42	1	**6**	3.42	1
	16	**89**	6.09	0.0178	**16**	9.48	1.0010	**16**	9.48	1.0012
	64	**161**	6.09	0.0048	**25**	10.58	1.0013	**25**	10.58	1.0012
	256	**298**	6.09	0.0012	**26**	10.81	1.0013	**25**	10.81	1.0017
	576	**437**	6.09	0.0005	**26**	10.86	1.0015	**25**	10.86	1.0018

Table 7.1

One element per subdomain ($H/h = 1$), BDDC and FETI-DP, $\rho_{ij} = 1$, random right hand side, rtol=10^{-7}.

		Unpreconditioned Schur Complement			BDDC			FETI-DP		
p	N	It.	λmax	λmin	It.	λmax	λmin	It.	λmax	λmin
2	4	**5**	6.41	1.4149	**2**	1.05	1	**2**	1.05	1
	16	**30**	20.32	0.3000	**7**	1.46	1.0006	**6**	1.46	1.0018
	64	**250**	203.12	0.0278	**9**	1.61	1.0009	**8**	1.61	1.0013
	256	**>1000**	2.03×10^4	0.0034	**9**	1.64	1.0011	**8**	1.62	1.0013
	576	**>1000**	2.03×10^6	0.0845	**9**	1.62	1.0008	**8**	1.63	1.0016
3	4	**9**	7.21	0.8299	**3**	1.21	1	**3**	1.21	1
	16	**42**	22.79	0.1781	**9**	2.10	1.0010	**8**	2.10	1.0004
	64	**374**	227.91	0.0165	**11**	2.27	1.0007	**11**	2.31	1.0006
	256	**>1000**	2.28×10^4	0.0046	**12**	2.34	1.0007	**12**	2.36	1.0004
	576	**>1000**	2.28×10^6	0.1094	**13**	2.36	1.0010	**13**	2.35	1.0006
4	4	**13**	7.47	0.5784	**3**	1.37	1	**3**	1.37	1
	16	**55**	23.63	0.1257	**10**	2.65	1.0010	**10**	2.65	1.0008
	64	**524**	236.30	0.0117	**13**	2.94	1.0022	**13**	2.94	1.0011
	256	**>1000**	2.36×10^4	0.0060	**14**	3.01	1.0004	**14**	3.00	1.0013
	576	**>1000**	2.36×10^6	0.1441	**15**	2.99	1.0005	**15**	3.00	1.0005
8	4	**22**	7.88	0.2545	**4**	1.89	1	**4**	1.89	1
	16	**86**	24.93	0.0570	**13**	4.37	1.0008	**12**	4.37	1.0004
	64	**917**	249.26	0.0053	**17**	4.86	1.0013	**18**	4.86	1.0009
	256	**>1000**	2.49×10^4	0.0074	**20**	4.93	1.0014	**19**	4.97	1.0008
	576	**>1000**	2.49×10^6	0.1978	**20**	4.96	1.0012	**20**	4.98	1.0009
16	4	**33**	8.22	0.1178	**5**	2.57	1	**5**	2.57	1
	16	**150**	25.99	0.0270	**15**	6.63	1.0010	**15**	6.63	1.0008
	64	**>1000**	259.89	0.0025	**21**	7.38	1.0012	**21**	7.38	1.0008
	256	**>1000**	2.60×10^4	0.0082	**26**	7.54	1.0006	**25**	7.53	1.0009
	576	**>1000**	2.60×10^6	0.2477	**26**	7.54	1.0010	**26**	7.55	1.0006
32	4	**52**	8.46	0.0564	**6**	3.42	1	**6**	3.42	1
	16	**230**	26.74	0.0131	**17**	9.44	1.0008	**17**	9.44	1.0009
	64	**>1000**	267.39	0.0012	**26**	10.52	1.0012	**25**	10.52	1.0009
	256	**>1000**	2.67×10^4	0.0092	**31**	10.72	1.0007	**31**	10.74	1.0008
	576	**>1000**	2.67×10^6	0.2878	**34**	10.78	1.0006	**33**	10.77	1.0005

Table 7.2

One element per subdomain ($H/h = 1$), BDDC and FETI-DP, $\rho_{ij} = 10^{(i-j)/4}$, random right hand side, rtol=10^{-7}.

		Unpreconditioned Schur Complement			BDDC		
p	N	It.	λmax	λmin	It.	λmax	λmin
2	4	**10**	5.36	0.5836	**3**	1.20	1.0000
	16	**22**	5.49	0.1680	**7**	2.06	1.0011
	64	**46**	5.51	0.0435	**10**	2.27	1.0012
	256	**89**	5.53	0.0110	**9**	2.33	1.0013
3	4	**14**	5.54	0.3669	**3**	1.43	1.0000
	16	**30**	5.60	0.1066	**9**	2.84	1.0005
	64	**58**	5.61	0.0028	**11**	3.16	1.0019
	256	**112**	5.61	0.0070	**12**	3.21	1.0011
4	4	**17**	5.64	0.2660	**4**	1.62	1.0000
	16	**35**	5.66	0.0780	**10**	3.51	1.0008
	64	**67**	5.67	0.0020	**13**	3.91	1.0019
	256	**130**	5.67	0.0051	**13**	4.00	1.0018
8	4	**26**	5.80	0.1254	**4**	2.22	1.0001
	16	**51**	5.81	0.0375	**12**	5.49	1.0024
	64	**96**	5.81	0.0098	**17**	6.12	1.0024
	256	**184**	5.81	0.0025	**17**	6.26	1.0028
16	4	**39**	5.95	0.0606	**5**	2.99	1.0001
	16	**77**	5.95	0.0184	**13**	8.02	1.0015
	64	**139**	5.95	0.0048	**20**	8.95	1.0019
	256	**268**	5.95	0.0012	**21**	9.16	1.0018
32	4	**58**	6.09	0.0297	**6**	3.91	1.0001
	16	**111**	6.09	0.0091	**15**	11.12	1.0014
	64	**201**	6.09	0.0024	**24**	12.41	1.0015
	256	**384**	6.09	0.0006	**24**	12.69	1.0021

Table 7.3

Four elements per subdomain ($H/h = 2$), BDDC, $\rho_{ij} = 1$, random right hand side, rtol=10^{-7}.

		Unpreconditioned Schur Complement			BDDC		
p	N	It.	λmax	λmin	It.	λmax	λmin
2	4	**13**	7.75	0.6199	**3**	1.20	1.0000
	16	**61**	31.01	0.1139	**8**	2.06	1.0013
	64	**830**	496.15	0.0069	**10**	2.27	1.0016
	256	**>1000**	1.27×10^5	0.0128	**11**	2.31	1.0009
3	4	**17**	8.04	0.3922	**3**	1.43	1.0000
	16	**78**	32.15	0.0722	**9**	2.84	1.0006
	64	**>1000**	514.41	0.0044	**12**	3.15	1.0014
	256	**>1000**	1.32×10^5	0.0174	**13**	3.20	1.0007
4	4	**22**	8.19	0.2856	**4**	1.62	1.0000
	16	**95**	32.76	0.1655	**9**	3.50	1.0001
	64	**>1000**	524.16	0.0032	**14**	3.89	1.0013
	256	**>1000**	1.34×10^5	0.0187	**16**	3.94	1.0009
8	4	**34**	8.54	0.1356	**4**	2.22	1.0001
	16	**161**	34.14	0.0252	**13**	5.47	1.0029
	64	**>1000**	546.27	0.0015	**17**	6.08	1.0022
	256	**>1000**	1.40×10^5	0.0252	**21**	6.21	1.0009
16	4	**52**	8.88	0.0657	**5**	2.99	1.0001
	16	**256**	35.51	0.0123	**14**	7.99	1.0015
	64	**>1000**	568.23	0.0008	**21**	8.89	1.0018
	256	**>1000**	1.45×10^5	0.0271	**27**	9.05	1.0011
32	4	**78**	9.13	0.0323	**6**	3.91	1.0001
	16	**371**	36.52	0.0061	**17**	11.07	1.0018
	64	**>1000**	584.37	0.0014	**25**	12.31	1.0016
	256	**>1000**	1.45×10^5	0.0321	**33**	12.50	1.0014

Table 7.4

Four elements per subdomain ($H/h = 2$), BDDC, $\rho_{ij} = 2^{i-j}$, random right hand side, rtol=10^{-7}.

			FETI-DP				irFETI-DP		
H/h	N	p	It.	λmax	λmin	Time (16 Proc.)	It.	Time (16 Proc.)	D.o.f.
1	4096	2	**7**	1.66	1.0074	2s	**7**	2s	16 129
		3	**9**	2.39	1.0017	2s	**8**	2s	36 481
		4	**10**	3.05	1.0217	4s	**9**	3s	65 025
		6	**12**	4.14	1.0098	4s	**11**	3s	146 689
		8	**13**	5.03	1.0067	6s	**11**	4s	261 121
		10	**14**	5.81	1.0157	8s	**12**	5s	408 321
		12	**15**	6.48	1.0260	11s	**13**	8s	588 289
		14	**15**	7.09	1.0267	15s	**13**	11s	801 025
		16	**16**	7.64	1.0121	23s	**14**	16s	1 046 529
		18	**16**	8.15	1.0107	33s	**14**	24s	1 324 801
		20	**17**	8.62	1.0114	53s	**14**	37s	1 635 841
		22	**17**	9.05	1.0123	75s	**15**	57s	1 979 649
		24	**18**	9.46	1.0138	94s	**16**	81s	2 356 225
		26	**18**	9.85	1.0160	115s	**16**	103s	2 765 569
		28	**18**	10.21	1.0183	155s	**16**	130s	3 207 681
		30	**18**	10.56	1.0206	200s	**17**	176s	3 682 561
		32	**19**	10.89	1.0227	256s	**17**	228s	4 190 209
2	1024	2	**9**	2.35	1.0020	1s	**8**	1s	16 129
		3	**11**	3.26	1.0167	1s	**11**	1s	36 481
		4	**12**	4.03	1.0146	2s	**11**	2s	65 025
		6	**14**	5.29	1.0098	2s	**12**	2s	146 689
		8	**15**	6.31	1.0232	4s	**12**	3s	261 121
		10	**16**	7.17	1.0304	6s	**14**	5s	408 321
		12	**17**	7.93	1.0177	10s	**15**	7s	588 289
		14	**18**	8.60	1.0118	17s	**16**	13s	801 025
		16	**18**	9.21	1.0133	23s	**17**	20s	1 046 529
		18	**19**	9.77	1.0154	30s	**17**	26s	1 324 801
		20	**19**	10.28	1.0186	43s	**17**	38s	1 635 841
		22	**19**	10.76	1.0218	60s	**18**	55s	1 979 649
		24	**20**	11.21	1.0247	83s	**18**	76s	2 356 225
		26	**21**	11.63	1.0271	106s	**18**	106s	2 765 569
		28	**21**	12.03	1.0294	164s	**18**	146s	3 207 681
		30	**21**	12.40	1.0309	219s	**18**	195s	3 682 561
		32	**22**	12.76	1.0230	276s	**18**	244s	4 190 209
4	256	2	**11**	3.18	1.0150	1s	**11**	1s	16 129
		3	**13**	4.26	1.0095	1s	**12**	1s	36 481
		4	**14**	5.14	1.0146	1s	**14**	1s	65 025
		6	**16**	6.56	1.0262	2s	**15**	2s	146 689
		8	**18**	7.70	1.0230	4s	**17**	4s	261 121
		10	**19**	8.65	1.0112	6s	**17**	6s	408 321
		12	**19**	9.49	1.0143	9s	**18**	9s	588 289
		14	**19**	10.22	1.0184	14s	**18**	14s	801 025
		16	**20**	10.89	1.0223	21s	**20**	20s	1 046 529
		18	**21**	11.49	1.0261	30s	**20**	29s	1 324 801
		20	**21**	12.05	1.0267	45s	**20**	42s	1 635 841
		22	**22**	12.57	1.0278	62s	**21**	59s	1 979 649
		24	**22**	13.05	1.0253	86s	**21**	84s	2 356 225
		26	**23**	13.51	1.0192	125s	**21**	116s	2 765 569
		28	**23**	13.94	1.0188	170s	**22**	164s	3 207 681
		30	**23**	14.34	1.0189	221s	**22**	219s	3 682 561
		32	**23**	14.73	1.0191	328s	**21**	280s	4 190 209

Table 7.5

Increasing polynomial degree ($p = 2, \ldots, 32$). Fixed subdomain sizes ($H/h = 1, 2, 4$). FETI-DP and inexact reduced FETI-DP (irFETI-DP, GMRES). Inexact reduced FETI-DP uses one iteration of the parallel algebraic multigrid solver BoomerAMG in hypre to precondition the coarse problem. We use BoomerAMG with parallel Gauss-Seidel smoothing.

FETI-DP							
p	N	H/h	It.	λmax	λmin	Time (16 Proc.)	D.o.f.
4	1024	1	**10**	3.04	1.0209	1s	16 129
		2	**12**	4.03	1.0146	2s	65 025
		4	**13**	5.17	1.0094	3s	261 121
		6	**15**	5.91	1.0182	5s	588 289
		8	**15**	6.46	1.0265	8s	1 046 529
		10	**16**	6.91	1.0295	13s	1 635 841
		12	**16**	7.29	1.0261	20s	2 356 225
		14	**16**	7.62	1.0183	25s	3 207 681
		16	**17**	7.91	1.0158	32s	4 190 209
		18	**17**	8.18	1.0144	40s	5 303 809
		20	**18**	8.41	1.0108	50s	6 548 481
		22	**18**	8.63	1.0112	63s	7 924 225
		24	**18**	8.83	1.0119	74s	9 431 041
		26	**18**	9.02	1.0126	87s	11 068 929
		28	**18**	9.20	1.0134	105s	12 837 889
		30	**18**	9.36	1.0142	125s	14 737 921
		32	**19**	9.52	1.0150	145s	16 769 025
8	256	1	**14**	5.00	1.0077	1s	16 129
		2	**16**	6.27	1.0216	2s	65 025
		4	**18**	7.70	1.0238	4s	261 121
		6	**19**	8.60	1.0111	8s	588 289
		8	**19**	9.28	1.0135	13s	1 046 529
		10	**19**	9.82	1.0162	20s	1 635 841
		12	**19**	10.28	1.0188	25s	2 356 225
		14	**19**	10.67	1.0211	35s	3 207 681
		16	**20**	11.02	1.0233	45s	4 190 209
		18	**21**	11.33	1.0251	59s	5 303 809
		20	**22**	11.61	1.0247	80s	6 548 481
		22	**22**	11.87	1.0253	100s	7 924 225
		24	**22**	12.10	1.0026	120s	9 431 041
		26	**22**	12.33	1.0258	140s	11 068 929
		28	**22**	12.53	1.0258	161s	12 837 889
		30	**22**	12.73	1.0251	193s	14 737 921
		32	**22**	12.91	1.0254	232s	16 769 025

Table 7.6
Increasing subdomain sizes ($H/h = 1, \ldots, 32$). Fixed polynomial degree ($p = 4, 8$).

	FETI-DP		irFETI-DP	
Proc.	It.	Time	It.	Time
2	22	337s	20	309s
4	22	172s	20	156s
8	22	89s	20	82s
16	22	45s	20	42s

Table 7.7: *Parallel scalability for p=20, N=256, H/h=4.*

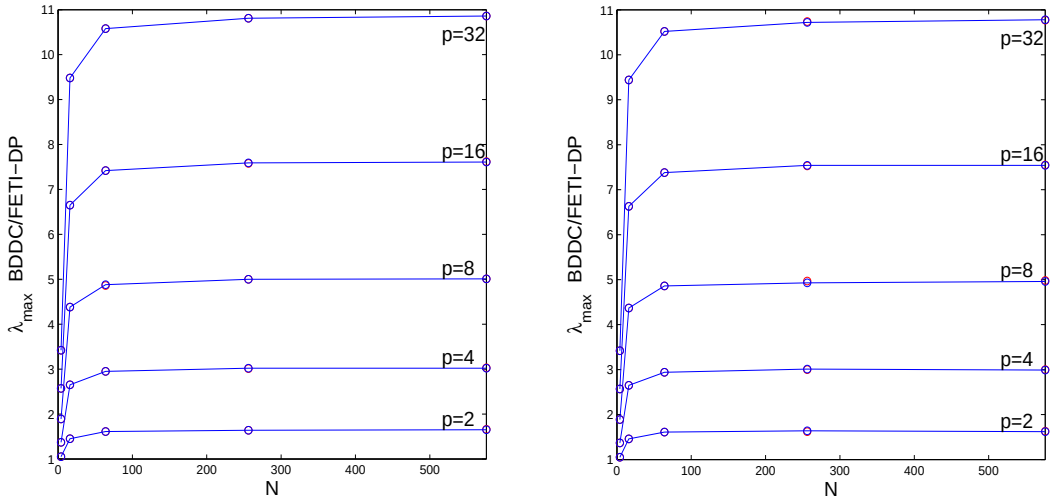

Figure 7.1
When do we reach asymptotic behavior for $H/h = 1$? Increasing number of subdomains; $\lambda_{\max}$ for BDDC (blue) and FETI-DP (red circles, barely visible). Upper: Homogeneous coefficient. Lower: Jumping coefficient $\rho_{ij} = 10^{(i-j)/4}$.

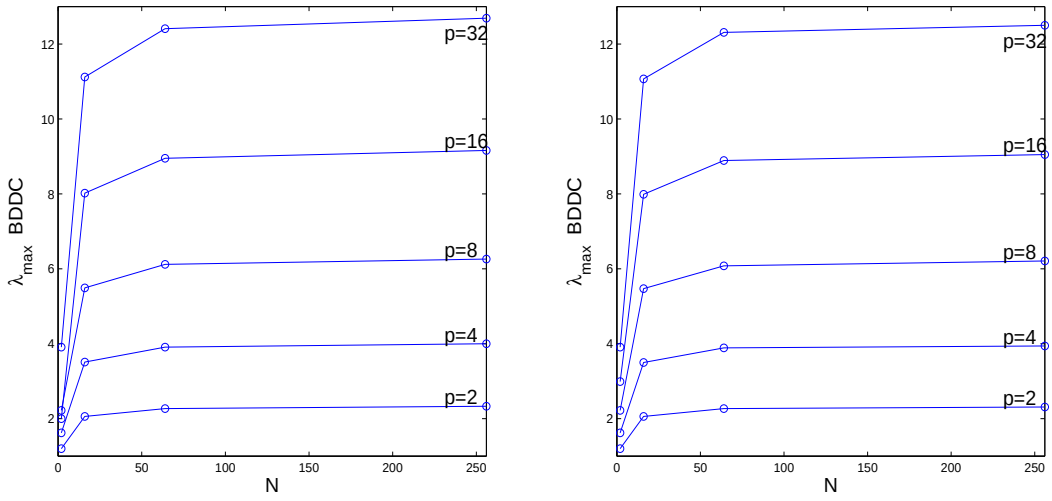

Figure 7.2
When do we reach asymptotic behavior for $H/h = 2$? Increasing number of subdomains; $\lambda_{\max}$ for BDDC. Upper: Homogeneous coefficient. Lower: Jumping coefficient $\rho_{ij} = 2^{(i-j)}$.

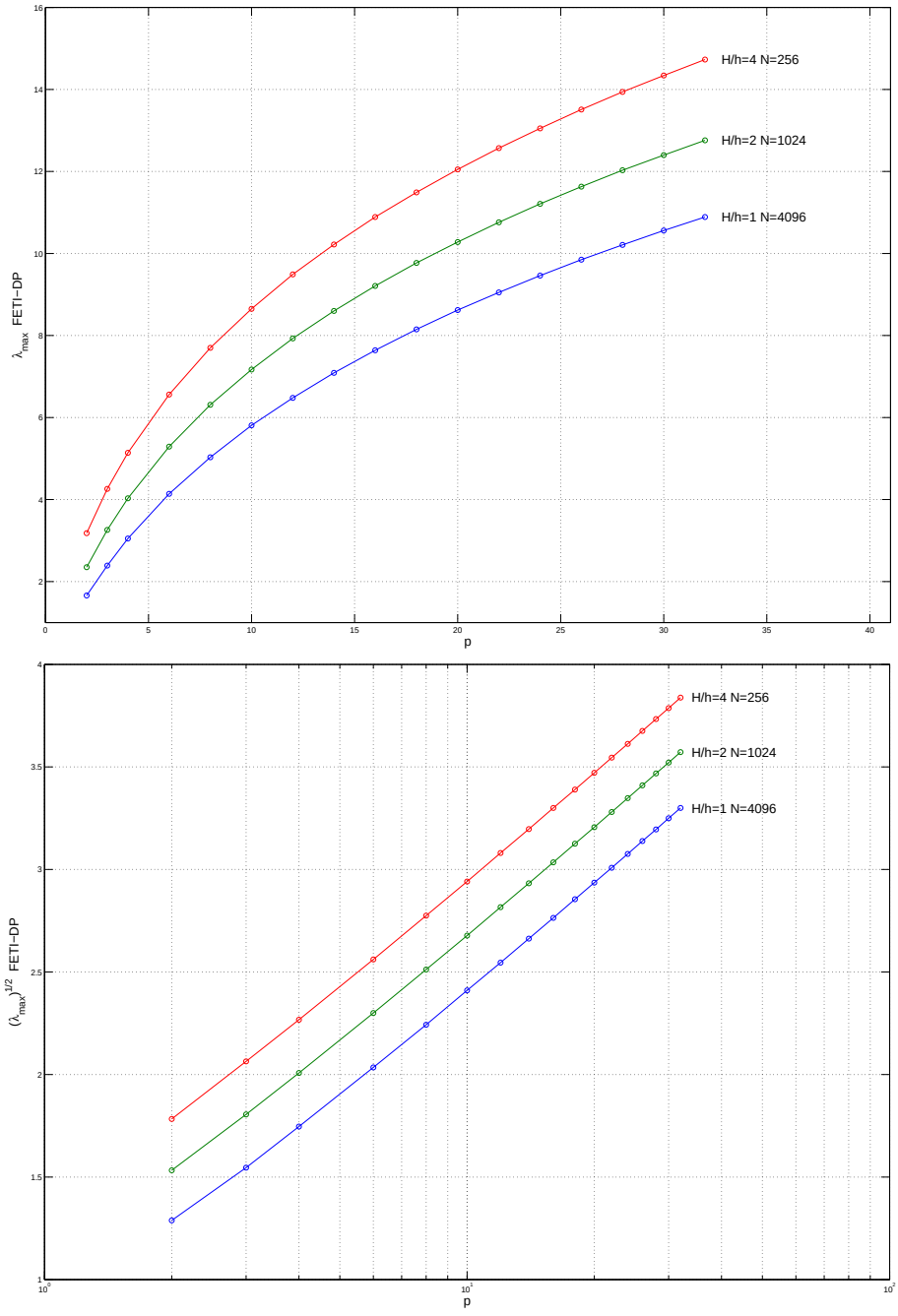

Figure 7.3

Increasing the polynomial degree, cf. Table 7.5. Largest eigenvalue $\lambda_{\max}$ *versus the polynomial degree (upper),* $\sqrt{\lambda_{\max}}$ *versus polynomial degree (lower).* $H/h = 1, 2, 4$*;* $N = 4096, 1024, 256$*; 16 129 – 4 190 209 d.o.f.; 2s – 280s using 16 processors.*

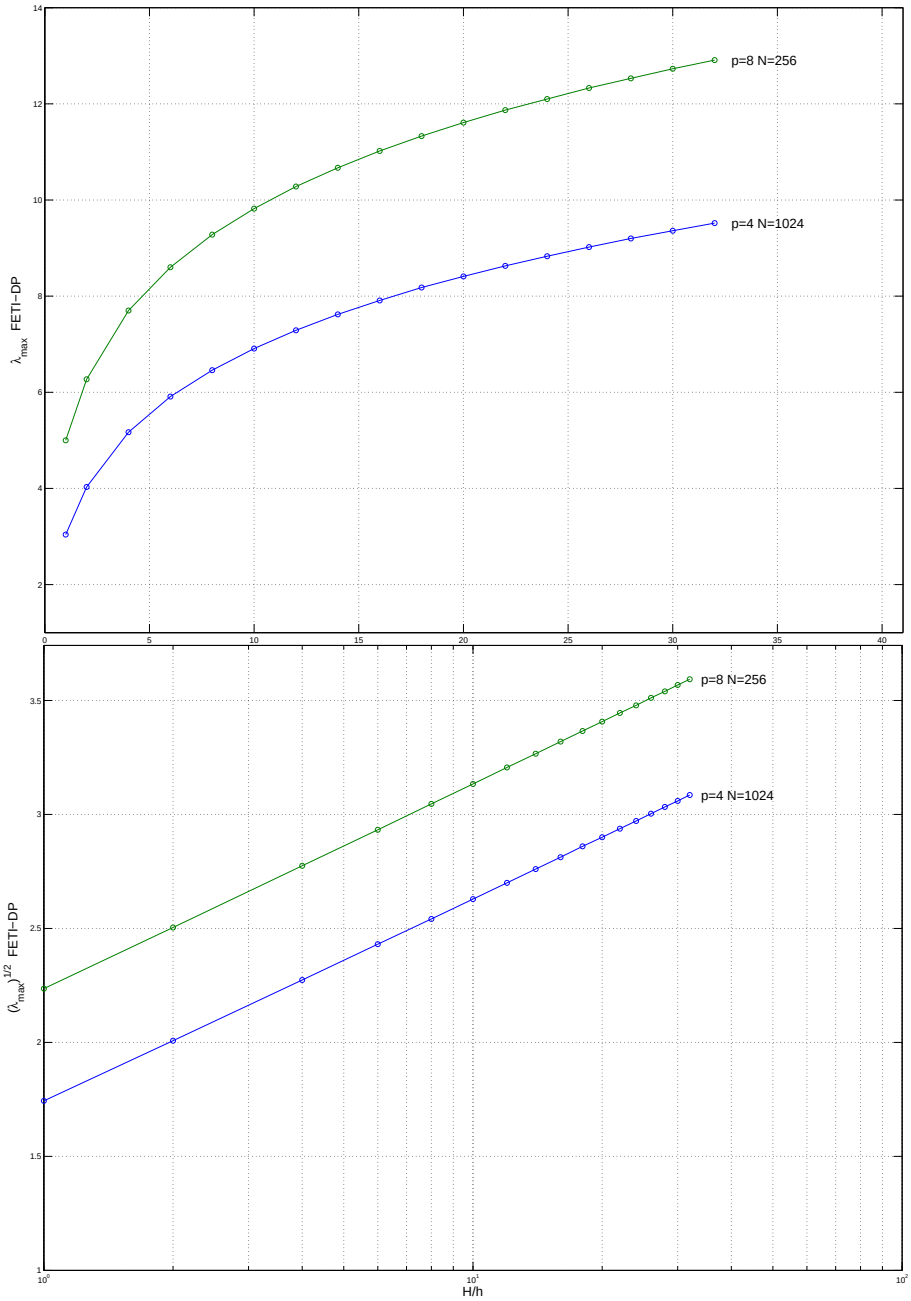

Figure 7.4

Increasing H/h for fixed polynomial degree, cf. Table 7.6. Largest eigenvalue $\lambda_{\max}$ versus H/h (upper), $\sqrt{\lambda_{\max}}$ versus H/h (lower). $p = 4, 8$; $N = 1024, 256$; 16 129 – 16 769 025 d.o.f.; 2s – 232s using 16 processors.

7.2 *hp* Finite Elements

In [134] and [133] Toselli and Vasseur develop, analyze and test Balancing Neumann-Neumann and the one-level FETI methods for *hp* finite element approximations. They focus on scalar elliptic problems on geometrically refined boundary layer meshes in 2D and later in 3D [135] and show a bound on the condition number that is polylogarithmic in the spectral degree and is also reflected in the numerical results presented in [134].

The problems in this section are taken from Toselli and Vasseur [134] who also provided their *hp* discretizations, see [134] for details on the meshes and discretizations.

In Problem IV from [134] a Laplace equation on the 2D unit square on an anisotropic mesh is considered. Anisotropic meshes can be applied to singularly perturbed problems although we do not consider them here. We have 3×3 subdomains, and a mesh refinement towards the boundary at $x = 0$ and $y = 0$. The number of layers n and the polynomial degree k is simultaneously increased. The results in Table 7.8 and 7.9 correspond to Table 10 (Neumann-Neumann) and Table 11 (one-level FETI) in [134], see also Figure 7.5. We apply our usual FETI-DP algorithm using vertex constraints. We find that the condition number of FETI-DP compares favorable to those of the Neumann-Neumann and one-level FETI and shows the same polylogarithmic behavior, see Figure 7.5.

In the second problem, Problem V from [134], the unit square is decomposed into 2×2 subdomains and the mesh is geometrically refined towards the interface of the subdomains at $x = 1/2$, $y = 1/2$. We also have a coefficient jump of 10^4 in a checkerboard distribution. Again, the number of layers n corresponds to the polynomial degree k. In contrast to the findings in [134] where linear growth in k was observed we observe a condition number of the FETI-DP operator nearly equal to one. However, the performance in terms of iterations counts is similar to the results obtained for the one-level FETI and the Neumann-Neumann method. It is well known for low order discretizations that the FETI-DP method performs exceptionally well for the special case of the 2D checker board problem.

All timings are given for a serial 2 GHz Intel Pentium IV personal computer using an early version of the FETI-DP implementation described in Chapter 3.

Problem IV from [134]:
Laplace problem, boundary layer mesh. 9 subdomains.
FETI-DP Algorithm A. Rtol = 10^{-14}.

k	It.	$\lambda_{\max}$	$\lambda_{\min}$	κ	Time
2	**9**	1.368	1.000	1.368	0.03s
3	**12**	1.924	1.001	1.922	0.02s
4	**14**	2.387	1.000	2.386	0.06s
5	**15**	2.815	1.000	2.814	0.13s
6	**15**	3.189	1.001	3.187	0.25s
7	**16**	3.537	1.001	3.534	0.57s
8	**17**	3.851	1.000	3.850	1.4s
9	**17**	4.146	1.000	4.144	2.3s
10	**18**	4.419	1.000	4.417	4.3s
11	**18**	4.677	1.001	4.674	7.2s
12	**18**	4.918	1.001	4.915	13s

Table 7.8
Laplace problem with 9 subdomains. With Dirichlet preconditioner.

Problem IV from [134]:
Laplace problem, boundary layer mesh. 9 subdomains.
FETI-DP Algorithm A. Rtol = 10^{-14}. No precon.

k	It.	$\lambda_{\max}$	$\lambda_{\min}$	κ	Time
2	**18**	1.870	0.509	3.674	0.04s
3	**30**	4.421	0.234	18.915	0.05s
4	**51**	7.263	0.123	59.289	0.16s
5	**72**	10.523	0.0656	160.478	0.38s
6	**107**	14.014	0.0350	400.474	0.95s
7	**153**	17.785	0.0185	962.121	2.2s
8	**226**	21.743	0.00964	2.254×10^3	7.8s
9	**296**	25.908	0.00524	4.940×10^3	16s
10	**429**	30.228	0.00254	11.892×10^4	35s
11	**497**	34.711	0.00132	26.325×10^4	68s
12	**500**	39.328	6.67032×10^{-4}	58.959×10^4	95s

Table 7.9
Laplace problem with 9 subdomains. Without preconditioner.

Problem V from [134]: Interface problem. 4 subdomains.
Checkerboard. $\rho_1 = 10^4$, $\rho_2 = 1$.
FETI-DP Algorithm A. Rtol $= 10^{-14}$.

k	It.	$\lambda_{\max}$	$\lambda_{\min}$	κ	Time
2	**3**	1.000	1.000	1.000	0.01s
3	**4**	1.001	1.000	1.001	0.02s
4	**4**	1.001	1.000	1.001	0.06s
5	**4**	1.001	1.000	1.001	0.16s
6	**4**	1.002	1.000	1.002	0.47s
7	**4**	1.002	1.000	1.002	1.11s
8	**4**	1.003	1.000	1.003	2.40s

Table 7.10
Interface problem with 4 subdomains. With Dirichlet preconditioner.

Problem V from [134]: Interface problem. 4 subdomains.
Checkerboard. $\rho_1 = 10^4$, $\rho_2 = 1$.
FETI-DP Algorithm A. Rtol $= 10^{-14}$. No precon.

k	It.	$\lambda_{\max}$	$\lambda_{\min}$	κ	Time
2	**10**	0.773	0.411	1.883	0.02s
3	**18**	1.664	0.370	4.500	0.03s
4	**25**	2.684	0.358	7.491	0.12s
5	**31**	3.845	0.353	10.899	0.27s
6	**39**	5.106	0.349	14.615	0.77s
7	**43**	6.464	0.347	18.630	2.16s
8	**49**	7.897	0.345	22.883	4.20s

Table 7.11
Interface problem with 4 subdomains. Without preconditioner.

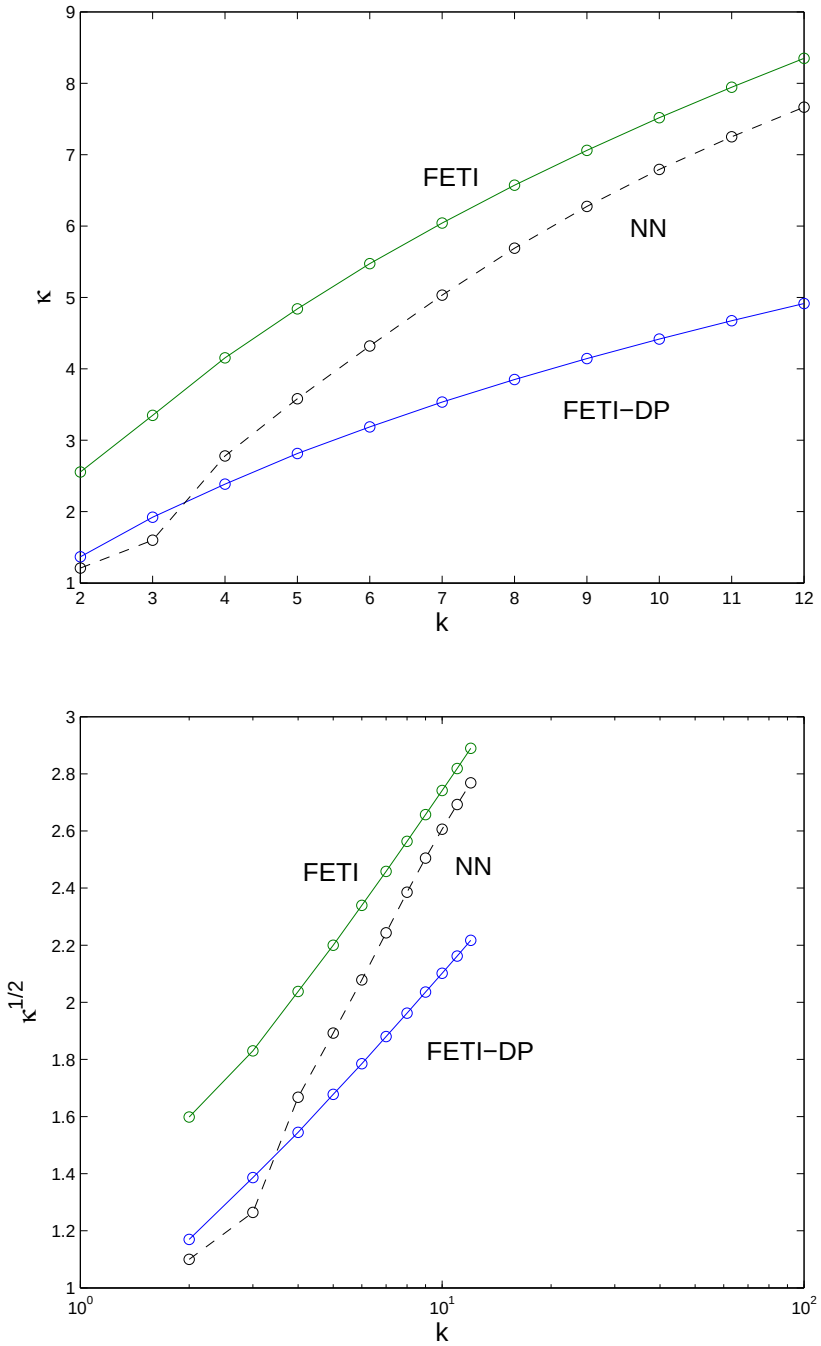

Figure 7.5

Upper diagram: Condition number for the boundary layer mesh (Problem IV) for FETI-DP (lowest). The values for Neumann-Neumann (middle) and one-level FETI (upper) are taken from [134].
Lower diagram: Square root of the condition number in a semilogarithmic plot.

Bibliography

[1] Mark F. Adams, Harun H. Bayraktar, Tony M. Keaveny, and Panayiotis Papadopoulos. Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. In *ACM/IEEE Proceedings of SC2004: High Performance Networking and Computing*, 2004.

[2] Martin Aigner. *Diskrete Mathematik.* Vieweg, 5th edition, 2004.

[3] Mark Ainsworth. A preconditioner based on domain decomposition for *hp*–FE approximation on quasi–uniform meshes. *SIAM J. Numer. Anal.*, 33:1358–1376, 1996.

[4] Patrick R. Amestoy, Iain S. Duff, and Jean-Yves L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods Appl. Mech. Engrg.*, 184:501–520, 2000.

[5] Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001.

[6] Patrick R. Amestoy, Abdou Guermouche, Jean-Yves L'Excellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Comput.*, 32(2):136–156, 2006.

[7] Cleve Ashcraft and Roger Grimes. SPOOLES: An object-oriented sparse matrix library. In *Proceedings of the 9th SIAM Conference on Parallel Processing for Scientific Computing 1999.*, March 1999.

[8] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[9] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matt Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc home page. 2001.

[10] Satish Balay, Victor Eijkhout, 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. Birkhäuser Press, 1997.

[11] Christine Bernardi and Yvon Maday. Polynomial interpolation results in Sobolev spaces. *J. Comput. Appl. Math.*, 43:53 – 80, 1992.

[12] Christine Bernardi and Yvon Maday. Spectral methods. In Philippe G. Ciarlet and Jacques Louis Lions, editors, *Handbook of Numerical Analysis, Vol. V*, pages 209–485. North-Holland, 1997.

[13] Manoj Bhardwaj, David Day, Charbel Farhat, Michel Lesoinne, Kendall H. Pierson, and Daniel Rixen. Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems. *Internat. J. Numer. Methods Engrg.*, 47:513–535, 2000.

[14] 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.

[15] Ion Bică. *Iterative Substructuring Algorithms for the p-version Finite Element Method for Elliptic Problems*. PhD thesis, Courant Institute of Mathematical Sciences, September 1997. TR-743, Department of Computer Science.

[16] Petter E. Bjørstad and Olof B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal., Vol. 23*, 6:1093–1120, 1986.

[17] James H. Bramble and Joseph E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Math. Comp.*, 50(181):1–17, 1988.

[18] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103–134, 1986.

[19] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2nd edition, 2002.

[20] Susanne C. Brenner and Li-yeng Sung. BDDC and FETI-DP without matrices and vectors. Technical report, University of South Carolina, 2005. http://www.math.sc.edu/ fem/papers/BDDC.html.

[21] 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.

[22] Claudio Canuto. Stabilization of spectral methods by finite element bubble functions. *Comput. Methods Appl. Mech. Engrg.*, 116(1-4):13–26, 1994.

[23] Claudio Canuto, M. Yousuff Hussaini, Alfio Quarteroni, and Thomas A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1988.

[24] Mario A. Casarin. Quasi-optimal Schwarz methods for the conforming spectral element discretization. *SIAM J. Numer. Anal.*, 34(6):2482–2502, 1997.

[25] Philippe G. Ciarlet. *Mathematical Elasticity Volume I: Three–Dimensional Elasticity*. North-Holland, 1988.

[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. Proceedings of the 14th International Conference on Domain Decomposition Methods in Science and Engineering; http://www.ddm.org/DD14.

[27] Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):165–195, June 2004.

[28] Michel O. Deville, Paul F. Fischer, and Ernest H. Mund. *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press, Cambridge, England, 2002.

[29] Clark R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[30] Clark R. Dohrmann. A substructuring preconditioner for nearly incompressible elasticity problems. Technical report, Sandia National Laboratories, Oct. 2004.

[31] Clark R. Dohrmann. An approximate BDDC preconditioner. Technical Report SAND 2005-5424, Sandia National Laboratories, August 2005.

[32] Clark R. Dohrmann and Richard B. Lehoucq. A primal based penalty preconditioner for elliptic saddle point systems. Technical Report SAND 2004-5964J, Sandia National Laboratories, 2004.

[33] Iain S. Duff and John K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Trans. Math. Software*, 9:302–325, 1983.

[34] Stanley C. Eisenstat, Howard C. Elman, and Martin H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 20 (2):345–357, 1983.

[35] Robert D. Falgout, Jim E. Jones, and Ulrike Meier Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. *Chapter in Numerical solution of Partial Differential Equations on Parallel Computers, A.M. Bruaset, P. Bjorstad, and A. Tveito, editors. Springer-Verlag, to appear.*, 2004. Also available as LLNL Technical Report UCRL-JRNL-205459, 2004.

[36] Robert D. Falgout, Jim E. Jones, and Ulrike Meier Yang. Pursuing scalability for hypre's conceptual interfaces. *ACM Trans. Math. Software*, 31:326–350, 2005. Also available as LLNL Technical Report UCRL-JRNL-205407.

[37] Robert D. Falgout and Ulrike Meier Yang. hypre: a library of high performance preconditioners. In P.M.A. Sloot, C.J.K. Tan., J.J. Dongarra, and A.G. Hoekstra, editors, *Computational Science - ICCS 2002*

Part III, volume 2331, pages 632–641. Lecture Notes in Computer Science, Springer-Verlag, 2002. Also available as LLNL Technical Report UCRL-JC-146175.

[38] Charbel Farhat, Po-Shu Chen, Franck Risler, and François-Xavier Roux. A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers. *Internat. J. Numer. Methods Engrg.*, 42:257–288, 1998.

[39] Charbel Farhat, Michel Lesoinne, Patrick Le Tallec, Kendall H. 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 Engrg.*, 50:1523–1544, 2001.

[40] Charbel Farhat, Michel Lesoinne, Patrick LeTallec, Kendall H. 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 Engrg.*, 50:1523–1544, 2001.

[41] Charbel Farhat, Michel Lesoinne, and Kendall H. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7:687–714, 2000.

[42] 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. *Comput. Methods Appl. Mech. Engrg.*, 155:129–152, 1998.

[43] Charbel Farhat, Jan Mandel, and François-Xavier Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:367–388, 1994.

[44] Charbel Farhat, Kendall H. 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. *Comput. Methods Appl. Mech. Engrg.*, 184:333–374, 2000.

[45] Charbel Farhat and François-Xavier Roux. A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32:1205–1227, 1991.

[46] Charbel Farhat and François-Xavier Roux. Implicit parallel processing in structural mechanics. In J. Tinsley Oden, editor, *Computational Mechanics Advances*, volume 2(1), pages 1–124. North-Holland, 1994.

[47] Eric Flauraud and Frédéric Nataf. Optimized interface conditions in domain decomposition methods. Applications at the semi-discrete and at the algebraic level to problems with extreme contrasts in the coefficients. Technical Report 524, CMAP, 2004.

[48] Pascal Frey. An interactive mesh visualization software. Technical Report RT-0253, INRIA, Institut de Recherche en Informatique et en Automatique, December 2001.

[49] Daniele Funaro. *Spectral Methods for Transport-Dominated Equations*. Springer-Verlag, Berlin, 1997.

[50] Alan George. Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, 10:345–363, 1973.

[51] Alan George and Joseph Liu. The evolution of the minimum degree ordering algorithm. *SIAM Rev.*, 31:1–19, 1989.

[52] Francis X. Giraldo and Timothy Warburton. A nodal triangle-based spectral element method for the shallow water equations on the sphere. *J. Comput. Phys.*, 207(1):129–150, 2005.

[53] Vivette Girault and Pierre-Arnaud Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer-Verlag, Berlin, 1986.

[54] Paulo Goldfeld. *Balancing Neumann-Neumann Preconditioners for Mixed Formulation of Almost-Incompressible Linear Elasticity*. PhD thesis, Courant Institute of Mathematical Sciences, September 2003. TR-847, Department of Computer Science.

[55] Paulo Goldfeld, Luca F. Pavarino, and Olof B. Widlund. Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. *Numer. Math.*, 95(2):283–324, 2003.

[56] Pierre Gosselet. *Méthodes de décomposition de domaine et méthodes d'accéleration pour les problèmes multichamps en non-linéaire*. PhD thesis, Université Paris 6, 2003.

[57] Anne Greenbaum. *Iterative methods for solving linear systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[58] Pierre Grisvard. *Elliptic Problems in Nonsmooth Domains*. Pitman, Boston, London, Melbourne, 1985.

[59] William Gropp and Ewing Lusk. *User's Guide for* `mpich`*, a Portable Implementation of MPI.* Mathematics and Computer Science Division, Argonne National Laboratory, 1996. ANL-96/6.

[60] William Gropp and Ewing Lusk. A high-performance MPI implementation on a shared-memory vector supercomputer. *Parallel Comput.*, 22(11):1513–1526, January 1997.

[61] William Gropp and Ewing Lusk. Sowing MPICH: A case study in the dissemination of a portable environment for parallel scientific computing. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):103–114, Summer 1997.

[62] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Comput.*, 22(6):789–828, September 1996.

[63] William Gropp, Anthony Skjellum, and Ewing Lusk. *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* MIT press, 2nd edition, 1999.

[64] Anshul Gupta. Recent advances in direct methods for solving unsymmetric sparse systems of linear equations. *ACM Trans. Math. Software*, 28(3), Sept. 2002.

[65] Bruce Hendrickson and Robert Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Sci. Stat. Comput.*, 16(2):452–469, 1995.

[66] Van E. Henson and Ulrike Meier Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41:155–177, 2002.

[67] Jim E. Jones and Panayot S. Vassilevski. AMGe based on element agglomerations. *SIAM J. Sci. Comput.*, 23:109–133, 2001. Also available as Lawrence Livermore National Laboratory Technical Report UCRL-JC-135441, August 1999.

[68] George E. Karniadakis and Spencer J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics.* Oxford University Press, 2nd edition, 2005.

[69] George Karypis and Vipin Kumar. METIS: Unstructured graph partitioning and sparse matrix ordering system. Version 2.0. Technical report, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455, August 1995.

[70] George Karypis, Kirk Schloegel, and Vipin Kumar. ParMETIS - Parallel graph partitioning and sparse matrix ordering. Version 3.1. Technical report, University of Minnesota, Department of Computer Science and Engineering, August 2003.

[71] Hyea H. Kim. A FETI-DP formulation of three dimensional elasticity problems with mortar discretization. Technical Report TR2005-863, Courant Institute of Mathematical Sciences, New York University, New York, USA, April 2005.

[72] Axel Klawonn. *Preconditioners for Indefinite Problems.* PhD thesis, Westfälische Wilhelms-Universität Münster, 1996.

[73] Axel Klawonn. Block-Triangular Preconditioners for Saddle Point Problems with a Penalty Term. *SIAM J. Sci. Comp.*, 19(1):172–184, January 1998.

[74] Axel Klawonn, Luca F. Pavarino, and Oliver Rheinbach. FETI-DP and BDDC domain decomposition methods for spectral elements in 2D. Technical report, 2006. In preparation.

[75] Axel Klawonn and Oliver Rheinbach. Inexact FETI-DP methods. Technical Report TR-SM-E-609, Department of Mathematics, University of Duisburg-Essen, Germany, July 2005. To appear in Internat. J. Numer. Methods Engrg.

[76] Axel Klawonn and Oliver Rheinbach. A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis. Technical Report TR-SM-E-601, Department of Mathematics, University of Duisburg-Essen, Germany, February 2005, revised November 2005. To appear in SIAM J. Sci. Comput.

[77] Axel Klawonn and Oliver Rheinbach. Robust FETI-DP methods for heterogeneous three dimensional linear elasticity problems. Technical Report TR-SM-E-607, Department of Mathematics, University of Duisburg-Essen, Germany, July 2005, revised January 2006. To appear in Comput. Methods Appl. Mech. Engrg.

[78] Axel Klawonn, Oliver Rheinbach, and Olof B. Widlund. Some computational results for Dual-Primal FETI methods for three dimensional elliptic problems. In Ralf Kornhuber, Ronald H.W. Hoppe, Jacques Périaux, Olivier Pironneau, Olof B. Widlund, and Jinchao Xu, editors, *Domain Decomposition Methods in Science and Engineering*. Springer-Verlag, Lecture Notes in Computational Science and Engineering, 2004. Proceedings of the 15th International Conference on Domain Decomposition Methods, Berlin, July 21-25, 2003.

[79] Axel Klawonn, Oliver Rheinbach, and Barbara I. Wohlmuth. Dual-primal iterative substructuring for almost incompressible elasticity. In David E. Keyes and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering*. Springer-Verlag, Lecture Notes in Computational Science and Engineering, 2006. Proceedings of the 16th International Conference on Domain Decomposition Methods, New York, NY, January 12-15, 2005. To appear.

[80] Axel Klawonn and Gerhard Starke. Block triangular preconditioners for nonsymmetric saddle point problems: Field-of-values analysis. *Numer. Math.*, 81:577–594, 1999.

[81] Axel Klawonn and Olof B. Widlund. A domain decomposition method with Lagrange multipliers and inexact solvers for linear elasticity. *SIAM J. Sci. Comput.*, 22(4):1199–1219, 2000.

[82] Axel Klawonn and Olof B. Widlund. Dual and dual-primal FETI methods for elliptic problems with discontinuous coefficients in three dimensions. In *Domain Decomposition Methods, Proceedings of the 12th International Conference on Domain Decomposition Methods, Chiba, Japan, October 1999*. DDM.org, 2001.

[83] Axel Klawonn and Olof B. Widlund. FETI and Neumann–Neumann iterative substructuring methods: Connections and new results. *Comm. Pure Appl. Math.*, 54:57–90, January 2001.

[84] Axel Klawonn and Olof B. Widlund. Dual-Primal FETI methods for linear elasticity. Technical Report 2004-855, Courant Institute of Mathematical Sciences, New York University, New York, USA, September, revised May 2006 2004.

[85] Axel Klawonn and Olof B. Widlund. Selecting constraints in Dual-Primal FETI methods for elasticity in three dimensions. In R. Kornhuber, R.H.W. Hoppe, D.E. Keyes, J. Périaux, O. Pironneau, and

J. Xu, editors, *Domain Decomposition Methods in Science and Engineering*, pages 67–81. Springer-Verlag, Lecture Notes in Computational Science and Engineering, 2005. Proceedings of the 15th International Conference on Domain Decomposition Methods, Berlin, July 21-25, 2003.

[86] 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, 159-179 2002.

[87] Axel Klawonn, Olof B. Widlund, and Maksymilian Dryja. Dual-Primal FETI methods with face constraints. In Luca F. Pavarino and Andrea Toselli, editors, *Recent developments in domain decomposition methods*, pages 27–40. Springer-Verlag, Lecture Notes in Computational Science and Engineering, Volume 23, 2002.

[88] Michel Lesoinne. A FETI-DP corner selection algorithm for three-dimensional problems. In Ismael Herrera, David E. Keyes, Olof B. Widlund, and Robert Yates, editors, *Domain Decomposition Methods in Science and Engineering. 14th International Conference on Domain Decomposition Methods*, pages 217–223, 2003. Cocoyoc in Morelos, Mexico, January 6–12.

[89] Jing Li. *Dual-Primal FETI methods for stationary Stokes and Navier-Stokes equations*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2002.

[90] Jing Li and Olof B. Widlund. BDDC algorithms for incompressible Stokes equations. Technical Report TR2005-861, Courant Institute of Mathematical Sciences, Department of Computer Science, April 2005.

[91] Jing Li and Olof B. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. Technical Report TR2005-871, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, July 2005. To appear in Comput. Methods Appl. Mech. Engrg.

[92] Jing Li and Olof B. Widlund. FETI–DP, BDDC, and Block Cholesky Methods. *Internat. J. Numer. Methods Engrg.*, 66(2):250–271, 2006.

[93] Xiaoye S. Li and James W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Math. Software*, 29(2):110–140, 2003.

[94] Richard J. Lipton and Robert E. Tarjan. A seperator theorem for planar graphs. *SIAM J. Appl. Math.*, 36:177–189, 1979.

[95] Frédéric Magoulès, François-Xavier Roux, and Stephanie Salmon. Optimal discrete transmission conditions for a non-overlapping domain decomposition method for the Helmholtz equation. *SIAM J. Sci. Comput.*, 25(5):1497–1515, 2004.

[96] Frédéric Magoulès, François-Xavier Roux, and Laurent Series. Algebraic way to derive absorbing boundary conditions for the Helmholtz equation. *J. Comput. Acoust.*, 13(3):433–454, 2005.

[97] 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.

[98] 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.

[99] Jan Mandel and Radek Tezaur. Convergence of a Substructuring Method with Lagrange Multipliers. *Numer. Math.*, 73:473–487, 1996.

[100] Jan Mandel and Radek Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

[101] Gary L. Miller, Shang-Hua Teng, William Thurston, and Stephen A. Vavasis. Automatic mesh partitioning. In Alan George, John R. Gilbert, and Joseph W. H. Liu, editors, *Graph Theory and Sparse Matrix Computations*. Springer-Verlag, 1993.

[102] Jindrich Nečas. *Les méthodes directes en théorie des équations elliptiques*. Academia, Prague, 1967.

[103] Richard Pasquetti, Luca F. Pavarino, Francesca Rapetti, and Elena Zampieri. Overlapping Schwarz preconditioners for Fekete spectral elements. In David E. Keyes and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering*. Springer-Verlag, Lecture Notes in Computational Science and Engineering, 2006. Proceedings of the 16th International Conference on Domain Decomposition Methods, New York, NY, January 12-15, 2005. To appear.

[104] Luca F. Pavarino. Neumann-Neumann algorithms for spectral elements in three dimensions. *RAIRO Mathematical Modelling and Numerical Analysis*, 31:471–493, 1997.

[105] Luca F. Pavarino and Timothy Warburton. Overlapping Schwarz methods for unstructured spectral elements. *J. Comput. Phys.*, 160 (1):298–317, 2000.

[106] Luca F. Pavarino and Olof B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, 2002.

[107] Kendall H. Pierson. *A family of domain decomposition methods for the massively parallel solution of computational mechanics problems.* PhD thesis, University of Colorado at Boulder, Aerospace Engineering, 2000.

[108] Kendall H. Pierson. Private communication, 2005.

[109] Kendall H. Pierson, Garth M. Reese, and Padma Raghavan. Experiences with FETI-DP in a production level finite element application. In Ismael Herrera, David E. Keyes, Olof B. Widlund, and Robert Yates, editors, *14th International Conference on Domain Decomposition Methods*, 2002.

[110] Garth Reese, Manoj Bhardwaj, Dan Segalman, Kenneth Alvin, and Brian Driessen. *Salinas User's Manual.* Sandia National Laboratories, 1998.

[111] Daniel Rixen and Charbel Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Internat. J. Numer. Methods Engrg.*, 44:489–516, 1999.

[112] John Ruge and Klaus Stüben. Efficient solution of finite difference and finite element equations by algebraic multigrid (ang). In J.D. Paddon and H. Holstein, editors, *The Institute of Mathematics and its Applications Conference Series*, volume 3, pages 169–212. Clarenden Press, Oxford, 1985.

[113] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, 1996.

[114] Yousef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[115] Olaf Schenk and Klaus Gärtner. Two-level dynamic scheduling in PARDISO: Improved scalability on shared memory multiprocessing systems. *Parallel Comput.*, 28(2):187–197, 2002.

[116] Joachim Schöberl. NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997.

[117] Joachim Schöberl. Multigrid methods for a parameter-dependent problem in primal variables. *Numer. Math.*, 84:97–119, 1999.

[118] Joachim Schöberl, Markus Melenk, Clemens Pechstein, and Sabine Zaglmayr. Additive Schwarz preconditioning for p-version triangular and tetrahedral finite elements. Technical Report Technical Report 2005-10, Johann Radon Institute for Computational and Applied Mathematics (RICAM), 2005.

[119] Christoph Schwab. *p- and hp-finite element methods. Theory and applications in solid and fluid mechanics.* Oxford University Press, New York, 1998.

[120] Robert Sedgewick. *Algorithms in C++.* Addison Wesley, 1992.

[121] Laurent Series, Frédéric Feyel, and François-Xavier Roux. Une méthode de décomposition de domaine à deux multiplicateurs de lagrange. Technical Report ONERA PRND 2004-49, ONERA, 2003.

[122] Spencer J. Sherwin and Mario A. Casarin. Low energy bases preconditioning for elliptic substructured solvers based on spectral/hp element discretizations. *J. Comput. Phys.*, 171:1–24, 2001.

[123] Jonathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Edition 1 1/4. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburg, PA, 15213, August 1994.

[124] Juan C. Simo and M.S. Rifai. A class of assumed strain methods and the method of incompatible modes. *Internat. J. Numer. Methods Engrg.*, 29:1595–1638, 1990.

[125] Valeria Simoncini. Block triangular preconditioners for symmetric saddle-point problems. *Applied Numerical Mathematics*, 49:63–80, 2004.

[126] Barry F. Smith, Petter Bjørstad, and William Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, New York, 1996.

[127] Barry F. Smith, Petter E. Bjørstad, and William Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[128] Gerhard Starke. Field-of-values-analysis of preconditioned iterative methods for nonsymmetric elliptic problems. *Numer. Math.*, 78:103–117, 1997.

[129] Klaus Stüben. An Introduction to Algebraic Multigrid. In *Multigrid* [137], pages 413–532. Also available as GMD Report 70, November 1999.

[130] Klaus Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128:281–309, 2001. Also available as GMD Report 69, November 1999.

[131] Klaus Stüben and Tanja Clees. *SAMG User's Manual, Release 22c (May 2005)*. Fraunhofer SCAI, Sankt Augustin, Germany, May 2005. `http://www.scai.fraunhofer.de/samg.htm`.

[132] Barna Szabó and Ivo Babuška. *Finite Element Analysis*. John Wiley & Sons, New York, 1991.

[133] Andrea Toselli and Xavier Vasseur. Neumann-Neumann and FETI preconditioners for *hp*-approximations on geometrically refined boundary layer meshes in two dimensions. Technical Report 02–15, Seminar für Angewandte Mathematik, ETH, Zürich, 2002. Submitted to Numer. Math.

[134] Andrea Toselli and Xavier Vasseur. A numerical study on Neumann-Neumann and FETI methods for *hp*-approximations on geometrically refined boundary layer meshes in two dimensions. *Comput. Methods Appl. Mech. Engrg.*, 192:4551–4579, 2003.

[135] Andrea Toselli and Xavier Vasseur. Domain decomposition preconditioners of Neumann-Neumann type for *hp*-approximations on boundary layer meshes in three dimensions. *IMA J. Numer. Anal.*, 24(1):123–156, 2004.

[136] Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin Heidelberg New York, 2005.

[137] Ulrich Trottenberg, Cornelis W. Oosterlee, and Anton Schüller. *Multigrid*. Academic Press, London, San Diego, 2001.

[138] Xuemin Tu. Three-level BDDC in two dimensions. Technical Report TR2004-856, Department of Computer Science, Courant Institute, New York University, 2004. To appear in Internat. J. Numer. Methods Engrg.

[139] Xuemin Tu. Three-level BDDC in three dimensions. Technical Report TR2005-862, Department of Computer Science, Courant Institute, New York University, 2005.

[140] Xuemin Tu. *BDDC Domain Decomposition Algorithms: Methods with Three Levels and for Flow in Porous Media*. PhD thesis, Courant Institute, New York University, January 2006. TR2005-879, Department of Computer Science, Courant Institute.

[141] Henry M. Tufo and Paul F. Fischer. Fast parallel direct solvers for coarse grid problems. Technical report, Brown University, 1997. To appear in J. Parallel & Distributed Computing.

[142] Panayot S. Vassilevski and Ludmil T. Zikatanov. Multiple vector preserving interpolation mappings in algebraic multigrid. Technical Report UCRL-JRNL-208036, Lawrence Livermore National Laboratory, November 2004. Submitted to SIAM J. Matrix Anal. Appl., 2004.

[143] Olof B. Widlund. An extension theorem for finite element spaces in continuum mechanics. In W. Hackbusch and K. Witsch, editors, *Notes on Numerical Fluid Mechanics*, volume 16, pages 110–122. Vieweg, Braunschweig, 1987. Proceedings of the Second GAMM-Seminar, Kiel, January 17 to 19, 1986.

[144] Christian Wieners. Robust multigrid methods for nearly incompressible elasticity. *Computing*, 64:289–306, 2000.