SCHRIFTENREIHE DES FACHBEREICHS MATHEMATIK

A Parallel Implementation of Dual-Primal FETI Methods for Three Dimensional Linear Elasticity Using a Transformation of Basis

by

Axel Klawonn and Oliver Rheinbach

SM-E-601 2005

Universität Duisburg-Essen

Eingegangen am 14.02.2005

A PARALLEL IMPLEMENTATION OF DUAL-PRIMAL FETI METHODS FOR THREE DIMENSIONAL LINEAR ELASTICITY USING A TRANSFORMATION OF BASIS

AXEL KLAWONN* AND OLIVER RHEINBACH†

February 14, 2005

Abstract. Dual-primal FETI methods for linear elasticity problems in three dimensions are considered. These are nonoverlapping domain decomposition methods where some *primal* continuity constraints across subdomain boundaries are required to hold throughout the iterations, whereas most of the constraints are enforced by Lagrange multipliers. An algorithmic framework for dual-primal FETI methods is described together with a transformation of basis to implement the primal constraints. Numerical results obtained from a parallel implementation of these algorithms applied to a model benchmark problem and to problems with more complicated geometries from industrial and biological applications are provided. These results show that the presented FETI-DP algorithms are numerical and parallel scalable.

Key words. domain decomposition, Lagrange multipliers, FETI, preconditioners, elliptic systems, elasticity, finite elements, parallel computing.

AMS subject classifications. 65F10, 65N30, 65N55

1. Introduction. 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 are all 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. 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 standard one-level 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.

FETI-DP algorithms were introduced by Farhat et al. in [9] for linear elasticity problems in the plane and then extended by Farhat, Lesoinne, and Pierson [10] to three dimensional elasticity problems. 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 [23]; 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 [13, 17, 18]; see also [26]. There, also a theory was provided which shows that the condition number in three dimensions can be again bounded polylogarithmically as a function of the dimension of the individual subdomain problems and that the bounds can otherwise be made independent of

*Fachbereich Mathematik, Universität Duisburg-Essen, Campus Essen, Universitätsstraße 3, 45117 Essen, Germany. E-mail: axel.klawonn@uni-essen.de, URL: http://www.uni-essen.de/numa.

†Fachbereich Mathematik, Universität Duisburg-Essen, Campus Essen, Universitätsstraße 3, 45117 Essen, Germany. E-mail: oliver.rheinbach@uni-essen.de.

the number of subdomains, the mesh size, and the jumps in the coefficients. More recently, new FETI-DP algorithms for three dimensional, linear elasticity problems were provided by Klawonn and Widlund [15, 16] 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.

The purpose of this article is to present results obtained from a parallel implementation of some of the algorithms developed in Klawonn and Widlund [15]. Here, we restrict ourselves to homogeneous elasticity problems without any jumps in the material coefficients; numerical experiments for heterogeneous problems are part of ongoing research. It is shown that using a transformation of basis for the implementation of the primal constraints leads to a parallel scalable domain decomposition method. We report on experiments for model benchmark problems and for more complicated geometries from industrial applications.

Strongly related to FETI-DP methods are the more recently developed Neumann-Neumann methods with constraints, also known as balancing domain decomposition methods by constraints (BDDC); cf. [8, 21, 22]. It was first shown in Mandel, Dohrmann, and Tezaur [22] that BDDC and FETI-DP have all non zero eigenvalues in common; see also Li and Widlund [20] for another approach. Neumann-Neumann methods with primal vertex constraints were also developed independently by Cros [6]; see also Fragakis and Papadrakakis [11] for further experimental work.

The remainder of this article is organized as follows. In Section 2, we introduce the equations of linear elasticity in three dimensions, the discretization by finite elements and the partition of the domain into substructures. In Section 3, we present the algorithmic framework for our FETI-DP methods and in Section 4 we describe how to perform the transformation of basis to implement the primal constraints. Finally, in Section 5 we present the numerical results obtained from a parallel implementation of our methods. First, we apply it to a benchmark problem on the unit cube, testing for numerical and parallel scalability, confirming the theoretical findings in [15]. We then show the parallel performance of our algorithms applied to geometries obtained from industrial and biological applications.

2. The equations of linear elasticity, finite elements, and geometry. The equations of linear elasticity model the displacement of a linear elastic material under the action of external and internal forces. The elastic body occupies a domain $\Omega \subset \mathbb{R}^3$, which is assumed to be polyhedral and of diameter one. We denote its boundary by $\partial\Omega$ and assume that one part of it, $\partial\Omega_D$, is clamped, i.e., with homogeneous Dirichlet boundary conditions, and that the rest, $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$, is subject to a surface force $\mathbf{g}$, i.e., a natural boundary condition. We can also introduce a body force $\mathbf{f}$, e.g., gravity. With $\mathbf{H^1}(\Omega) := (H^1(\Omega))^3$, the appropriate space for a variational formulation is the Sobolev space $\mathbf{H^1_0}(\Omega, \partial\Omega_D) := \{\mathbf{v} \in \mathbf{H^1}(\Omega) : \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}$. The linear elasticity problem consists in finding the displacement $\mathbf{u} \in \mathbf{H^1_0}(\Omega, \partial\Omega_D)$ of the elastic body Ω, such that

$$\int_\Omega G(\mathbf{x})\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x} + \int_\Omega G(\mathbf{x})\,\beta(\mathbf{x}) \operatorname{div}\mathbf{u} \operatorname{div}\mathbf{v}\, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v}\rangle \quad \forall \mathbf{v} \in \mathbf{H^1_0}(\Omega, \partial\Omega_D). \tag{2.1}$$

Here G and β are material parameters which depend on the Young's modulus $E > 0$ and the Poisson ratio $\nu \in (0, 1/2]$; we have $G = E/(1+\nu)$ and $\beta = \nu/(1-2\nu)$. In this article, we only consider the case of compressible elasticity, which means that the Poisson ratio ν is bounded away from $1/2$. Furthermore, $\varepsilon_{ij}(\mathbf{u}) := \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ is the linearized strain tensor, and

$$\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) = \sum_{i,j=1}^{3} \varepsilon_{ij}(\mathbf{u})\varepsilon_{ij}(\mathbf{v}), \quad \langle \mathbf{F}, \mathbf{v} \rangle := \int_\Omega \mathbf{f}^T \mathbf{v}\, d\mathbf{x} \; + \; \int_{\partial\Omega_N} \mathbf{g}^T \mathbf{v}\, d\sigma.$$

For convenience, we also introduce the notation

$$(\varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}))_{L_2(\Omega)} := \int_\Omega \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x}.$$

The bilinear form associated with linear elasticity is then

$$a(\mathbf{u}, \mathbf{v}) = (G\,\varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}))_{L_2(\Omega)} \; + \; (G\,\beta \operatorname{div} \mathbf{u}, \operatorname{div} \mathbf{v})_{L_2(\Omega)}.$$

The wellposedness of the linear system (2.1) follows immediately from the continuity and ellipticity of the bilinear form $a(\cdot, \cdot)$, where the first follows from elementary inequalities and the latter from Korn's first inequality; see, e.g., Ciarlet [5].

The null space $\mathbf{ker}\,(\varepsilon)$ of ε is the space of rigid body motions which is spanned by the three translations

$$\mathbf{r}_1 := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{r}_2 := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{r}_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{2.2}$$

and the three rotations

$$\mathbf{r}_4 := \begin{bmatrix} x_2 - \hat{x}_2 \\ -x_1 + \hat{x}_1 \\ 0 \end{bmatrix}, \mathbf{r}_5 := \begin{bmatrix} -x_3 + \hat{x}_3 \\ 0 \\ x_1 - \hat{x}_1 \end{bmatrix}, \mathbf{r}_6 := \begin{bmatrix} 0 \\ x_3 - \hat{x}_3 \\ -x_2 + \hat{x}_2 \end{bmatrix}. \tag{2.3}$$

Here $\hat{\mathbf{x}} \in \Omega$ to shift the origin to a point in Ω.

We will only consider compressible elastic materials. It is therefore sufficient to discretize our elliptic problem (2.1) by low order, conforming finite elements, e.g., linear or trilinear elements.

Let us assume that a triangulation τ^h of Ω is given which is shape regular and has a typical diameter of h. We denote by $\mathbf{W}^h := \mathbf{W}^h(\Omega) \subset \mathbf{H_0^1}(\Omega, \partial\Omega_D)$ the corresponding conforming finite element space of finite element functions. The corresponding discrete problem is then

$$a(\mathbf{u}_h, \mathbf{v}_h) = \langle \mathbf{F}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathbf{W}^h. \tag{2.4}$$

When there is no risk of confusion, we will drop the subscript h.

Let the domain $\Omega \subset \mathbb{R}^3$ be decomposed into nonoverlapping subdomains $\Omega_i, i = 1, \ldots, N$, each of which is the union of finite elements with matching finite element nodes on the boundaries of neighboring subdomains across the interface Γ. The interface Γ is the union of three different groups of open sets, namely, subdomain faces, edges, and vertices. We denote individual faces, edges, and vertices by $\mathcal{F}, \mathcal{E}$, and $\mathcal{V}$, respectively. To define faces, edges, and vertices, we introduce certain equivalence

classes. Let us denote the sets of nodes on $\partial\Omega$, $\partial\Omega_i$, and Γ by $\partial\Omega_h$, $\partial\Omega_{i,h}$, and Γ_h, respectively. For any interface nodal point $x \in \Gamma_h$, we define

$$\mathcal{N}_x := \{j \in \{1, \ldots, N\} : x \in \partial\Omega_j\},$$

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

Associated with the nodes of the finite element mesh, we have a graph, the nodal graph, which represents the node-to-node adjacency. For a given node $x \in \Gamma_h$, we denote by $\mathcal{C}_{con}(x)$ the connected component of the nodal subgraph, defined by $\mathcal{N}_x$, to which x belongs. For two interface points $x, y \in \Gamma_h$, we introduce an equivalence relation by

$$x \sim y :\Longleftrightarrow \mathcal{N}_x = \mathcal{N}_y \text{ and } y \in \mathcal{C}_{con}(x).$$

We can now describe faces, edges, and vertices using their equivalence classes. Here, $|G|$ denotes the cardinality of the set G. We define

DEFINITION 2.1.

$$\begin{array}{rcl} x \in \mathcal{F} & :\Longleftrightarrow & |\mathcal{N}_x| = 2 \\ x \in \mathcal{E} & :\Longleftrightarrow & |\mathcal{N}_x| \geq 3 \ \textit{ and } \ \exists y \in \Gamma_h, y \neq x, \ \textit{such that } y \sim x \\ x \in \mathcal{V} & :\Longleftrightarrow & |\mathcal{N}_x| \geq 3 \ \textit{ and } \ \nexists y \in \Gamma_h, \ \textit{such that } x \sim y. \end{array}$$

In the case of a decomposition into regular substructures, e.g., cubes or tetrahedrons, our definition of faces, edges, and vertices is conform with our basic geometric intuition; see Figure 2.1. On the other hand, for subdomains generated by an automatic mesh partitioner, the situation can be quite complicated. We can, e.g., have several edges with the same index set $\mathcal{N}_x$ or an edge and a vertex with the same $\mathcal{N}_x$. In practice, we can also have situations when there are not enough edges and potential edge constraints for some subdomains. Then, we have to use constraints on some extra edges on $\partial\Omega_N$, which otherwise would be regarded as part of a face. A similar problem might occur for flat structures for which additional constraints might be required for each subdomain. Therefore, we introduce an alternative definition of edges.

DEFINITION 2.2. *An edge is the largest connected set of nodes with the same index set N_x where $|N_x| \geq 3$ or $|N_x| \geq 2$ and x is on $\partial\Omega_N$.*

If needed, we will increase the number of edges in unstructured cases by switching locally from the definition of edges given in Definition 2.1 to Definition 2.2 by splitting edges into several edges.

3. The FETI-DP algorithms. In this section, we introduce a matrix formulation of the FETI-DP algorithms considered in this article. For each subdomain Ω_i, $i = 1, \ldots, N$, we assemble local stiffness matrices K_i and local load vectors $\mathbf{f}_i$ of the elasticity problem (2.4). Furthermore, we denote by $\mathbf{u}_i$ the local solution vectors of nodal values. In the dual-primal FETI methods, we distinguish between dual and primal displacement variables according to the way the continuity of the solution in those variables is established. Dual displacement variables are those, for which the continuity is enforced by a continuity constraint and Lagrange multipliers $\boldsymbol{\lambda}$ and thus, continuity is not established until convergence of the iterative method is reached, as in the classical one-level FETI methods. On the other hand, continuity of the primal displacement variables is enforced explicitly in each iteration step by subassembly of the local stiffness matrices K_i at the primal displacement variables. This subassembly yields a symmetric, positive definite stiffness matrix $\widetilde{K}$ which is not block diagonal

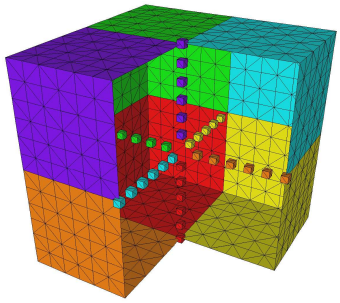

FIG. 2.1. *For $8 = 2 \times 2 \times 2$ subdomains the 6 edges in the interior of Ω are shown (two subdomains in front of the cube are not shown).*

anymore but coupled at the primal displacement variables. Let us note that this coupling yields a global problem which is necessary to obtain a parallel and numerical scalable algorithm. An obvious choice of primal variables are (selected) vertices for which a partial assembly is then carried out. We will see in Section 4 that there are other possible choices, such as edge averages over selected edges; cf., Klawonn and Widlund [15] for other possible choices. If a transformation of basis is used, then the edge average constraints can be treated algorithmically exactly as primal vertices; cf. Section 4.

We will use subscripts I, Δ, and Π, to denote the interior, dual, and primal displacement variables, respectively, and obtain for the local stiffness matrices, load vectors, and solution vectors of nodal values

$$K^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} & K_{\Pi I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} & K_{\Pi\Delta}^{(i)T} \\ K_{\Pi I}^{(i)} & K_{\Pi\Delta}^{(i)} & K_{\Pi\Pi}^{(i)} \end{bmatrix}, \mathbf{u}^{(i)} = \begin{bmatrix} \mathbf{u}_I^{(i)} \\ \mathbf{u}_\Delta^{(i)} \\ \mathbf{u}_\Pi^{(i)} \end{bmatrix}, \mathbf{f}^{(i)} = \begin{bmatrix} \mathbf{f}_I^{(i)} \\ \mathbf{f}_\Delta^{(i)} \\ \mathbf{f}_\Pi^{(i)} \end{bmatrix}.$$

By subassembly in the primal displacement variables, we obtain

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

where a tilde indicates the subassembled matrices and where

$$K_{BB} = \begin{bmatrix} K_{BB}^{(1)} & O & \cdots & O \\ O & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & K_{BB}^{(N)} \end{bmatrix}, \quad K_{BB}^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} \end{bmatrix},$$

and

$$\widetilde{K}_{\Pi B} = [\widetilde{K}_{\Pi B}^{(1)} \cdots \widetilde{K}_{\Pi B}^{(N)}].$$

Introducing local assembly operators $R_\Pi^{(i)}$ which map from the local primal displace-

ment variables $\mathbf{u}_\Pi^{(i)}$ to the global, assembled $\widetilde{\mathbf{u}}_\Pi$, we have

$$\widetilde{K}_{\Pi B}^{(i)} = R_\Pi^{(i)} K_{\Pi B}^{(i)}, \quad i = 1, \ldots, N, \qquad \widetilde{\mathbf{u}}_\Pi = \sum_{i=1}^{N} R_\Pi^{(i)} \mathbf{u}_\Pi^{(i)},$$

and

$$\widetilde{K}_{\Pi\Pi} = \sum_{i=1}^{N} R_\Pi^{(i)} K_{\Pi\Pi}^{(i)} R_\Pi^{(i)T}.$$

We also introduce the notation

$$\mathbf{u}_B = [\mathbf{u}_I \ \mathbf{u}_\Delta]^T, \mathbf{f}_B = [\mathbf{f}_I \ \mathbf{f}_\Delta]^T, \mathbf{u}_B^{(i)} = [\mathbf{u}_I^{(i)} \ \mathbf{u}_\Delta^{(i)}]^T, \text{ and } \mathbf{f}_B^{(i)} = [\mathbf{f}_I^{(i)} \ \mathbf{f}_\Delta^{(i)}]^T.$$

To guarantee continuity at the dual displacement variables, we introduce a jump operator B, which is constructed from $\{0, 1, -1\}$, in such a way that the values of the solution $\mathbf{u}_\Delta$, associated with more than one subdomain, coincide when $B\mathbf{u}_B = 0$; the interior variables $\mathbf{u}_I$ remain unchanged and thus the corresponding entries in B are zero. These constraints are very simple and just express that the nodal values coincide across the interface; clearly, we do not need any constraints associated with the primal displacement variables. However, we will otherwise use all possible constraints and thus work with a fully redundant set of Lagrange multipliers as in [14, Section 5]; cf. also [24]. Thus, for an edge node common to four subdomains, we will use six constraints rather than choosing as few as three.

We can now reformulate the finite element discretization of (2.4) as

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & O \\ B & O & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_B \\ \widetilde{\mathbf{u}}_\Pi \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_B \\ \widetilde{\mathbf{f}}_\Pi \\ \mathbf{0} \end{bmatrix}$$

Elimination of the interior and dual displacement variables $\mathbf{u}_B$ and of the primal displacement variables $\widetilde{\mathbf{u}}_\Pi$ leads to

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B^T \\ O & \widetilde{S}_{\Pi\Pi} & -\widetilde{S}_{\Lambda\Pi}^T \\ O & O & -F \end{bmatrix} \begin{bmatrix} \mathbf{u}_B \\ \widetilde{\mathbf{u}}_\Pi \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_B \\ \widetilde{\mathbf{f}}_\Pi - \widetilde{K}_{\Pi B} K_{BB}^{-1} \mathbf{f}_B \\ -\mathbf{d} \end{bmatrix}$$

with

$$\begin{aligned} \widetilde{S}_{\Pi\Pi} &= \widetilde{K}_{\Pi\Pi} - \widetilde{K}_{\Pi B} K_{BB}^{-1} \widetilde{K}_{\Pi B}^T, \qquad \widetilde{S}_{\Lambda\Pi}^T = \widetilde{K}_{\Pi B} K_{BB}^{-1} B^T, \\ F &= B K_{BB}^{-1} B^T + B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \widetilde{S}_{\Pi\Pi}^{-1} \widetilde{K}_{\Pi B} K_{BB}^{-1} B^T, \\ \mathbf{d} &= B K_{BB}^{-1} \mathbf{f}_B + B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \widetilde{S}_{\Pi\Pi}^{-1} \left(\widetilde{\mathbf{f}}_\Pi - \widetilde{K}_{\Pi B} K_{BB}^{-1} \right). \end{aligned}$$

Since K_{BB} is a block-diagonal matrix, we obtain

$$\begin{aligned} F &= \sum_{i=1}^{N} B^{(i)} \left(K_{BB}^{(i)} \right)^{-1} B^{(i)T} \\ &+ \left(\sum_{i}^{N} B^{(i)} (K_{BB}^{(i)})^{-1} \widetilde{K}_{\Pi B}^{(i)T} \right) \widetilde{S}_{\Pi\Pi}^{-1} \left(\sum_{j}^{N} \widetilde{K}_{\Pi B}^{(j)} (K_{BB}^{(j)})^{-1} B^{(j)T} \right). \end{aligned} \tag{3.1}$$

We see that in a matrix-vector multiplication with F, we only need to solve local linear systems in the first sum. To apply the second term in (3.1) to a vector, we first apply the right sum in it to that vector; this represents a gather operation. Next, we have to solve a linear system with the matrix $\widetilde{S}_{\Pi\Pi}$, which yields our global problem neeeded for scalability. Finally, we apply the left sum of the second term in (3.1) to the result; this represents a scatter operation. A similar form can be derived for the right hand side $\mathbf{d}$.

To define the FETI-DP Dirichlet preconditioner M^{-1}, we first need to introduce a scaled jump operator B_D; this is done by scaling the contributions of B associated with the dual displacement variables from individual subdomains. We can write

$$B_D = [B_D^{(1)}, \ldots, B_D^{(N)}],$$

where the $B_D^{(i)}$ are defined as follows: each row of $B^{(i)}$ with a nonzero entry corresponds to a Lagrange multiplier connecting the subdomain Ω_i with a neighboring subdomain Ω_j at a point $x \in \partial\Omega_{i,h} \cap \partial\Omega_{j,h}$. We obtain $B_D^{(i)}$ by multiplying each such row of $B^{(i)}$ with $G_j(x)/\sum_{k\in\mathcal{N}_x} G_k(x)$ where $G_l(x) := G(x)$ for $x \in \partial\Omega_l, l = 1, \ldots, N$. In the homogeneous case, this results in scaling each such row with the reciproke of the multiplicity $|\mathcal{N}_x|$ of the node $x \in \partial\Omega_{i,h} \cap \partial\Omega_{j,h}$; thus it is also called multiplicity scaling.

Let us now consider the matrix representation of our preconditioner M^{-1}. For this, we need certain Schur complement matrices $S^{(i)}, i = 1, \ldots, N$. Each $S^{(i)}$ is obtained from the local stiffness matrices $K^{(i)}$ by eliminating the interior displacement variables. Denoting the displacement variables on the local interface $\partial\Omega_i$ by the subscript Γ, we then have

$$K^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Gamma I}^{(i)T} \\ K_{\Gamma I}^{(i)} & K_{\Gamma\Gamma}^{(i)} \end{bmatrix}$$

and the local Schur complements $S^{(i)}$ are obtained as

$$S^{(i)} = K_{\Gamma\Gamma}^{(i)} - K_{\Gamma I}^{(i)}(K_{II}^{(i)})^{-1}K_{\Gamma I}^{(i)T}.$$

We now define the block-diagonal Schur complement matrix S as

$$S = \begin{bmatrix} S^{(1)} & & O \\ & \ddots & \\ O & & S^{(N)} \end{bmatrix}.$$

Our preconditioner is then given in matrix form by

$$M^{-1} = B_D R_\Gamma^T S R_\Gamma B_D^T = \sum_{i=1}^{N} B_D^{(i)} R_\Gamma^{(i)T} S^{(i)} R_\Gamma^{(i)} B_D^{(i)T}. \tag{3.2}$$

Here, $R_\Gamma^{(i)}$ are restriction matrices that restrict the degrees of freedom of a subdomain to its interface and $R_\Gamma = \mathrm{diag}_i(R_\Gamma^{(i)})$.

Let us note that generally the local Schur complements $S^{(i)}$ are never built explicitly, instead, in each iteration step and each application of M^{-1} to a vector, local linear systems with $K^{(i)}$ are solved. This corresponds to the solution of local Dirichlet problems, thus, this preconditioner is also known as the Dirichlet preconditioner.

Clearly, a matrix-vector multiplication with M^{-1} requires only the solution of local linear systems and can be carried out completely in parallel.

The FETI-DP method can now be described as the pcg method for solving the preconditioned linear system

$$M^{-1}F\boldsymbol{\lambda} = M^{-1}\mathbf{d}$$

and the algorithm is given in Algorithm 3.1.

ALGORITHM 3.1.

(i) Initialization: $\mathbf{r}^0 := \mathbf{d} - F\boldsymbol{\lambda}^0$

(ii) Iterate for $k = 1, 2, \ldots$, *until convergence,*

$$\begin{aligned}
\mathbf{z}^{k-1} &:= M^{-1}\mathbf{r}^{k-1} \\
\beta^k &:= \frac{\langle \mathbf{z}^{k-1}, \mathbf{r}^{k-1}\rangle}{\langle \mathbf{z}^{k-2}, \mathbf{r}^{k-2}\rangle} \quad [\beta^1 := 0] \\
\mathbf{p}^k &:= \mathbf{z}^{k-1} + \beta^k \mathbf{p}^{k-1} \quad [\mathbf{p}^1 := \mathbf{z}^0] \\
\alpha^k &:= \frac{\langle \mathbf{z}^{k-1}, \mathbf{r}^{k-1}\rangle}{\langle \mathbf{p}^k, F\mathbf{p}^k\rangle} \\
\boldsymbol{\lambda}^k &:= \boldsymbol{\lambda}^{k-1} + \alpha^k \mathbf{p}^k \\
\mathbf{r}^k &:= \mathbf{r}^{k-1} - \alpha^k F\mathbf{p}^k
\end{aligned}$$

4. Primal constraints and a change of basis. We have to decide how to choose the primal displacement variables. The simplest choice is to choose them as certain selected primal vertices of the subdomains, see Farhat et al. [9], where this approach was first considered. In this section, we show that edge average constraints can be treated the same way by using a transformation of basis. In the following, we will denote the FETI-DP algorithm which exclusively uses selected vertices as primal displacement constraints as Algorithm A, cf. Klawonn, Widlund, and Dryja [17], where this notation was introduced. In Lesoinne [19], an algorithm for the choice of selected vertices as primal variables is suggested in the context of choosing a relatively small number of vertices to guarantee invertibility of the subproblems and otherwise enhance the coarse problem by face averages using optional Lagrange multipliers.

For scalar, second order, elliptic differential equations in three dimensions, certain edge and face averages were suggested in [13], [17] as additional primal constraints. For linear elasticity problems in three dimensions, Farhat, Lesoinne, and Pierson [10] considered face averages as primal constraints. In Klawonn and Widlund [15], edge averages and first order edge moments were introduced and analyzed in order to obtain scalable algorithms which are also robust for arbitrarily large jumps of the material coefficients across the subdomain interface. The algorithms described in this article are based on [15].

There are two different approaches to implement the edge and face constraints, one using optional Lagrange multipliers, which form a part of the global, coarse problem, cf. [10] and [15, Sect. 6.1], and the other using a change of basis; cf. [15, Sect. 6.2]. The latter approach generally leads to smaller and computationally more efficient coarse problems. Using this approach, the invertibility of the local problems and the positive definiteness of the entire problem can also be guaranteed without any vertex constraints. In fact, vertex constraints are only needed for problems with very challenging distributions of the material coefficients; see [15, Sect. 8.3].

We now describe the approach using an explicit change of basis. As a result, the finite element functions associated with dual displacement vectors will have zero

edge averages over primal edges. In addition, we introduce these averages as primal variables. Here and in the following, we define an edge as primal if at least one component of the displacements has the same edge average across the interface on this edge.

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 original nodal finite element space. Denoting the edge unknowns in the new basis by $\widehat{\mathbf{u}}_E$, we have

$$\mathbf{u}_E = T_E \widehat{\mathbf{u}}_E.$$

Such a transformation matrix T_E can be constructed separately for each edge with three primal edge constraints. Ordering the three edge averages last, a possible implementation of T_E is

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

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

$$T^{(i)} = \begin{bmatrix} I & O & O \\ O & I & O \\ O & O & T_E^{(i)} \end{bmatrix}.$$

Here, we assume that the variables are ordered interior variables first, interface variables not related to the primal edges second, and the variables on the primal edges last, i.e., a typical vector of nodal unknowns is of the form $[\mathbf{u}_I^{(i)T}, \mathbf{u}_{\overline{\Gamma}}^{(i)T}, \mathbf{u}_E^{(i)T}]^T$. Here, we denote the interface variables not related to a primal edge by the subscript $\overline{\Gamma}$. We note that $T_E^{(i)}$ is a 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 $\mathbf{u}^{(i)} = T^{(i)}\widehat{\mathbf{u}}^{(i)}$, we obtain

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

where the upper left 2×2 block matrix is not affected by the basis transformation. The primal variables in the new basis consist now of averages but we note that we

might also have selected primal vertices as additional primal variables. The primal variables belonging to Ω_i are denoted by $\mathbf{u}_\Pi^{(i)}$ and the remaining, dual displacement variables by $\mathbf{u}_\Delta^{(i)}$. By construction, the basis functions associated with the new dual displacement variables have zero edge average over primal edges.

In the same manner the indices Δ_E and Π_E indicate the dual and primal displacement variables associated with the primal edges constraints. Denoting the transformed matrices by an overline and ordering the primal edge variables last, we obtain

$$T^{(i)T}K^{(i)}T^{(i)} = \left[\begin{array}{cc|cc} K_{II}^{(i)} & K_{I\overline{\Gamma}}^{(i)} & \overline{K}_{I\Delta_E}^{(i)} & \overline{K}_{I\Pi_E}^{(i)} \\ K_{\overline{\Gamma}I}^{(i)} & K_{\overline{\Gamma}\overline{\Gamma}}^{(i)} & \overline{K}_{\overline{\Gamma}\Delta_E}^{(i)} & \overline{K}_{\overline{\Gamma}\Pi_E}^{(i)} \\ \hline \overline{K}_{\Delta_E I}^{(i)} & \overline{K}_{\Delta_E\overline{\Gamma}}^{(i)} & \overline{K}_{\Delta_E\Delta_E}^{(i)} & \overline{K}_{\Delta_E\Pi_E}^{(i)} \\ \overline{K}_{\Pi_E I}^{(i)} & \overline{K}_{\Pi_E\overline{\Gamma}}^{(i)} & \overline{K}_{\Pi_E\Delta_E}^{(i)} & \overline{K}_{\Pi_E\Pi_E}^{(i)} \end{array}\right].$$

Denoting the primal vertices by a subscript Π_V and the remaining dual displacement variables by a subscript Δ, we can then write $\mathbf{u}_{\overline{\Gamma}}^{(i)} = [\mathbf{u}_\Delta^{(i)T}\ \mathbf{u}_{\Pi_V}^{(i)T}]^T$. Using this splitting for the local stiffness matrices $K^{(i)}$ accordingly, ordering the primal variables $\mathbf{u}_{\Pi_V}^{(i)}$ and $\mathbf{u}_{\Pi_E}^{(i)}$ last, and combining them as primal variables $\mathbf{u}_\Pi^{(i)} = [\mathbf{u}_{\Pi_V}^{(i)T}, \mathbf{u}_{\Pi_E}^{(i)T}]^T$, we obtain

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

Assembling the primal contributions of each transformed $K^{(i)}$ and ordering the primal variables last, we obtain

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

In our FETI-DP algorithm described in Section 3, 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

$$\left[\begin{array}{ccc} K_{BB} & \widetilde{K}_{\Pi B}^T & B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & O \\ B & O & O \end{array}\right] \left[\begin{array}{c} \mathbf{u}_B \\ \tilde{\mathbf{u}}_\Pi \\ \boldsymbol{\lambda} \end{array}\right] = \left[\begin{array}{c} \mathbf{f}_B \\ \tilde{\mathbf{f}}_\Pi \\ \mathbf{0} \end{array}\right].$$

We note that, after the change of basis has been carried out, we can always use the same implementation as for Algorithm A since the algorithmic description in

Section 3 does not depend on a specific choice of primal and dual variables. Note that the explicit transformation of basis introduces small dense blocks into the system matrix but the local problems as well as the Schur complement $\tilde{S}_{\Pi\Pi}$ remain symmetric positive definite.

The following condition number estimate for FETI-DP algorithms using edge averages on selected edges and/or selected vertices as primal constraints can be deduced from Klawonn and Widlund [15, Theorem 1].

THEOREM 4.1. *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 from the results in Klawonn and Widlund [15] that primal vertices are not needed to obtain a good condition number bound. We thus introduce a new notation and denote the algorithm using edge averages on selected primal edges as the only primal constraints by **Algorithm D_E**.

We note that in [15] a variant of Algorithm D_E is considered which in addition uses first order moments on selected edges and possibly selected primal vertices. In this case, a condition number estimate is shown which is also independent of discontinuities of the material coefficients across the interface.

5. Numerical results. We apply our new implementation of FETI-DP using a basis transformation 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 Subsections 5.1, 5.2, 5.3. The domains considered in the second and third subsection are decomposed in irregularly shaped substructures by using ParMetis [12]. In all problems, we use linear tetrahedral finite elements and, for simplicity, a Young modulus of $E = 210$ and a Poisson 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 selected 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 obtained by Definition 2.1 primal; additionally, in Subsections 5.2 and 5.3, we add some edges obtained by Definition 2.2. All computations of Subsections 5.1 and 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 Xeon processors with 1 or 2 GByte of memory each and uses a Myrinet connection. The numerical results given in Table 5.1, Figure 5.6 and in Subsection 5.3 have been carried out on a 16 processor computing cluster in Essen with eight dual 2.2 GHz Opteron nodes and 4 GByte of memory for each processor. This cluster uses a Gigabit ethernet connection.

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 5.1 and 5.2. 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 5.3 and Figure 5.1. 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 5.1 and 5.2, the cube is partitioned into smaller cubes with $H/h = 14$ which results in 8 332 d.o.f. for each subdomain. In Tables 5.1 and 5.2, 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 [3], [2], [4] for solving the coarse problem and the local problems; cf. Table 5.1. Next,

Proc.	N	$1/h$	d.o.f.	Coarse	$\lambda_{\min}$	$\lambda_{\max}$	Iter	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 5.1
Results for a cube divided into smaller cubes with a fixed subdomain size of 8 332 *d.o.f. using PETSc as subdomain and coarse grid sparse direct solver.*

we present a set of experiments using the sparse direct solver package MUMPS 4.3.2 [1] together with the optimized BLAS libraries for the Xeon architecture to solve the coarse problem and the local problems; cf. Table 5.2. Both set of experiments show

Proc.	N	$1/h$	d.o.f.	Coarse	$\lambda_{\min}$	$\lambda_{\max}$	Iter	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 5.2
Results for a cube divided into smaller cubes with a fixed subdomain size of 8 332 *d.o.f. using MUMPS as subdomain and coarse grid sparse direct solver.*

that our FETI-DP algorithm using only edge averages as 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 $N = 4 \times 4 \times 4 = 64$ subdomains. We keep $H = 1/4$ fixed and vary H/h between 4 and 24. Algorithm D_E has a coarse problem size of 324. The growth of the number of iterations and of the largest eigenvalue of Algorithms D_E is shown in Figure 5.1.

d.o.f.	H/h	Iter	$\lambda_{\min}$	$\lambda_{\max}$
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 5.3

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}

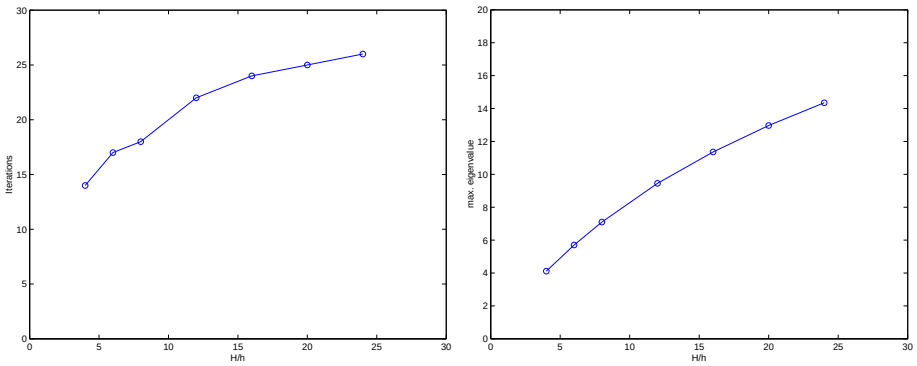

FIG. 5.1. *Growth of number of iterations (left) and λ_{max} (right) of Algorithm D_E for the unit cube with variable H/h and $4 \times 4 \times 4$ subdomains. .*

All of the numerical results in this subsection confirm the theoretical condition number estimate given in Theorem 4.1; cf. also [15, Theorem 1].

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.

The direct subdomain solves as well as the coarse problem solve are performed by using again 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 5.2, 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 partioned 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 eigenvalue are $\lambda_{\min} = 1.02$ and $\lambda_{\max} = 19.63$, respectively. The parallel scalability results on 2 to 16 processors are given in Table 5.4.

The second problem in this subsection, mechanical part B, cf. Figure 5.3, is discretized by 581 394 linear tetrahedral finite elements resulting in a global number of

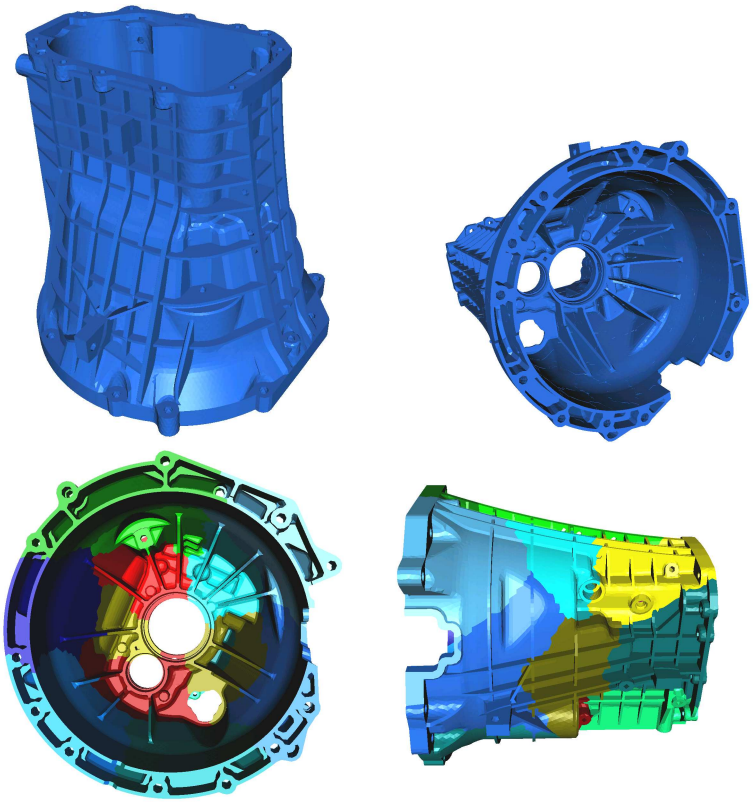

FIG. 5.2. *Mechanical part A; courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany*

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

TABLE 5.4

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

380 709 d.o.f. It is partioned into $N = 64$ subdomains using ParMetis. In all of the experiments reported in Table 5.5, 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. 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 5.5.

The third problem in this subsection, mechanical part C, cf. Figure 5.4, is discretized by 1 291 933 linear tetrahedral finite elements yielding a global number of 841 836 d.o.f. It is partioned into $N = 64$ subdomains using ParMetis. In all of the experiments reported in Table 5.6, the number of iterations is 32 for a relative reduction of the preconditioned dual residual by 10^{-7}. The size of the coarse problem is 957 d.o.f. The smallest and largest eigenvalue are $\lambda_{\min} = 1.03$ and $\lambda_{\max} = 27.77$,

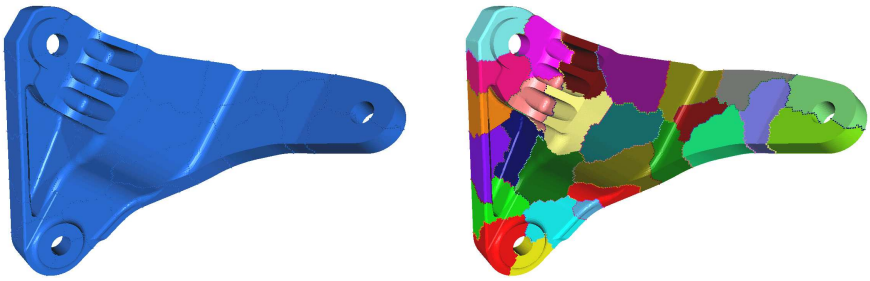

FIG. 5.3. *Mechanical part B; courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany*

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

TABLE 5.5
Parallel scalability results for mechanical part B: CPU times for FETI-DP using MUMPS as local and coarse sparse direct solver and for MUMPS applied to the undecomposed problem.

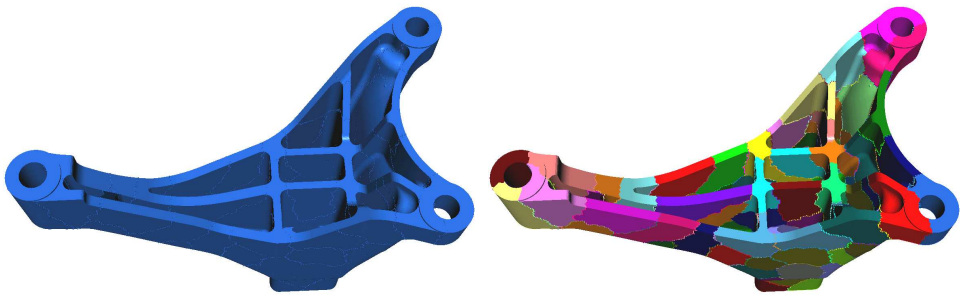

FIG. 5.4. *Mechanical part C; courtesy of GETRAG FORD Transmissions GmbH, Cologne, Germany*

respectively. The parallel scalability results on 16 to 64 processors are given in Table 5.6.

Proc.	16	32	64
FETI-DP D_E/MUMPS	**49s**	**26s**	**17s**
MUMPS	failed	1 177s	889s

TABLE 5.6
Parallel scalability results for mechanical part C: CPU times for FETI-DP with 64 subdomains using MUMPS as local and coarse sparse direct solver and for MUMPS applied to the undecomposed problem.

In Table 5.7 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 5.7
Parallel scalability results for mechanical part C: CPU times for FETI-DP with 512 subdomains using MUMPS as local and coarse sparse direct solver.

From our numerical experiments we see that the FETI-DP algorithms using edge averages only 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.

5.3. A cancellous bone geometry. In this subsection, we present results of our FETI-DP algorithm applied to a linearly elastic domain which has 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 [25] to produce a volume mesh from it. It is partitioned into $N = 96$ subdomains using ParMetis. In all of our experiments reported in Table 5.8, 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 eigenvalue are $\lambda_{\min} = 1.02$ and $\lambda_{\max} = 43.57$, respectively. The parallel scalability results on 1 to 16 processors are given in Table 5.8. As sparse direct solvers for the coarse and the local problems, we use UMFPACK 4.3 [7].

In Figure 5.6 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 $\|F\boldsymbol{\lambda}^n - \mathbf{d}\|_2/\|F\boldsymbol{\lambda}^0 - \mathbf{d}\|_2$ explicitly in each step; cf. Figure 5.6. Let us note that the true residual is only used in the experiments for the results shown in Figure 5.6. 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.

Acknowledgment. The authors gratefully acknowledge 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. The authors also gratefully acknowledge 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. The sec-

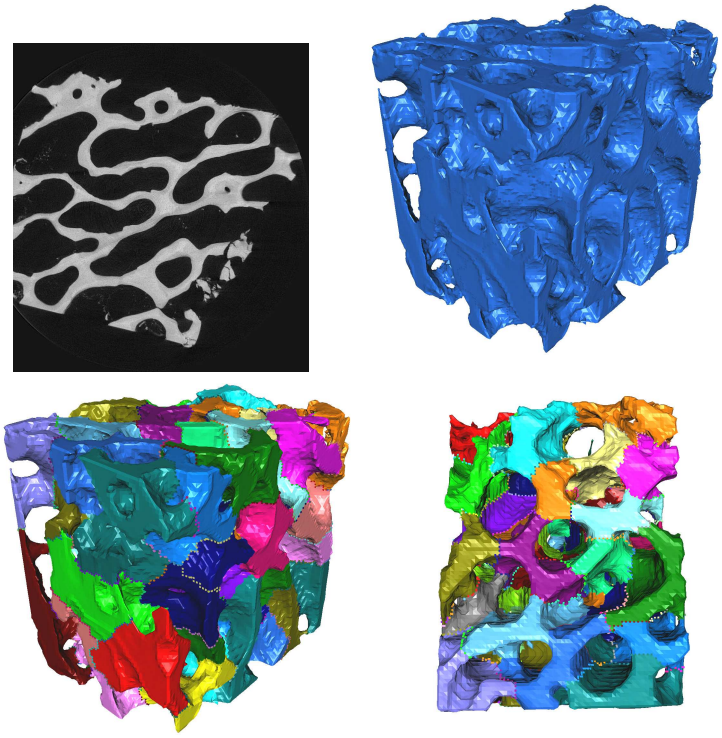

FIG. 5.5. *Cancellous bone. Upper left: Cross section of X-ray Computer Tomography; Upper right: Finite element discretization with* 907 609 *tetrahedrons and* 620 730 *d.o.f. Lower left and right: Different views of a decomposition of the bone into subdomains by ParMetis.*

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

TABLE 5.8
Parallel scalability of FETI-DP for cancellous bone geometry.

ond author acknowledges many fruitful discussions with Barry Smith and Matthew Knepley while visiting the Mathematics and Computer Science Division at Argonne National Laboratory, USA. The authors wish to thank Hans-Jürgen Philipsenburg, Holger Hein and GETRAG FORD Transmissions GmbH, Cologne, Germany. Last but not least, the authors have had many interesting discussions with Olof Widlund, Courant Institute of Mathematical Sciences, USA.

REFERENCES

[1] Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[2] 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,

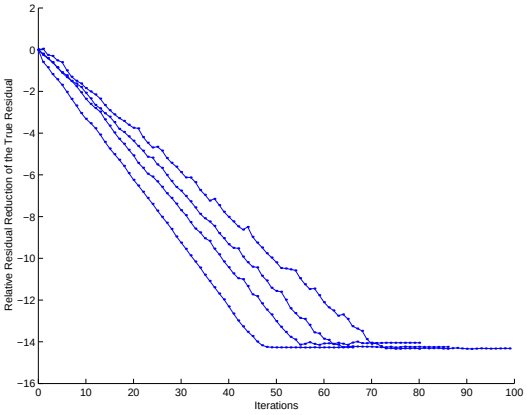

FIG. 5.6. *History of relative residual reductions of the true residual. From left to right: cube benchmark problem (105^3), mechanical part A, mechanical part B, cancellous bone geometry.*

2004.

[3] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matt Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc home page. http://www.mcs.anl.gov/petsc, 2005.

[4] 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.

[5] Philippe G. Ciarlet. *Mathematical Elasticity Volume I: Three–Dimensional Elasticity*. North–Holland, 1988.

[6] 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.

[7] Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, June 2004.

[8] Clark R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci.Comput.*, 25(1):246–258, 2003.

[9] Charbel Farhat, Michel Lesoinne, Patrick LeTallec, Kendall Pierson, and Daniel Rixen. FETI-DP: A dual-primal unified FETI method - part i: A faster alternative to the two-level FETI method. *Int. J. Numer. Meth. Engrg.*, 50:1523–1544, 2001.

[10] Charbel Farhat, Michel Lesoinne, and Kendall Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[11] Yannis Fragakis and Manolis Papadrakakis. The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods. *Comput. Methods Appl. Mech. Engrg*, 192:3799–3830, 2003.

[12] 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.

[13] 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.

[14] 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.

[15] 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 2004.

[16] 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.

[17] 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.

[18] 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.

[19] 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. Fourteenth International Conference on Domain Decomposition Methods*, pages 217–223, 2003. Cocoyoc in Morelos, Mexico, January 6–12.

[20] Jing Li and Olof B. Widlund. FETI-DP, BDDC, and Block Cholesky Methods. Technical Report TR2004-857, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, December 2004. http://cs.nyu.edu/csweb/Research/TechReports/TR2004-857/TR2004-857.pdf.

[21] Jan Mandel and Clark R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Lin. Alg. Appl.*, 10:639–659, 2003.

[22] Jan Mandel, Clark R. Dohrmann, and Radek Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 2004. Sixth IMACS International Symposium on Iterative Methods in Scientific Computing, Denver, Colorado, USA, 2003. To appear.

[23] Jan Mandel and Radek Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

[24] Daniel Rixen and Charbel Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Int. J. Numer. Meth. Engng.*, 44:489–516, 1999.

[25] Joachim Schöberl. Netgen - an advancing front 2d/3d-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997.

[26] 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.