Adaptive Domain Decomposition Methods for Finite and Boundary Element Equations *

by
G. Haase, B. Heise, M. Kuhn and U. Langer

(Johannes Kepler University Linz)

Abstract

The use of the FEM and BEM in different subdomains of a non–overlapping Domain Decomposition (DD) and their coupling over the coupling boundaries (interfaces) brings about several advantages in many practical applications. The paper presents parallel solvers for large-scaled coupled FE–BE–DD equations approximating linear and nonlinear plane magnetic field problems as well as plane linear elasticity problems. The parallel algorithms presented are of asymptotically optimal, or, at least, almost optimal complexity and of high parallel efficiency.

Key words: Linear and nonlinear elliptic boundary value problems, magnetic field problems, elasticity problems, domain decomposition, finite elements, boundary elements, coupling, solvers, preconditioners, parallel algorithms

AMS (MOS) subject classification: 65N55, 65N22, 65F10, 65N30, 65N38, 65Y05, 65Y10

1 Introduction

The Domain Decomposition (DD) approach offers many opportunities to marry the advantages of the Finite Element Method (FEM) to those of the Boundary Element Method (BEM) in many practical applications. For instance, in the magnetic field computation for electric motors, we can use the BEM in the air subdomains including the exterior of the motor more successfully than the FEM which is prefered in ferromagnetic materials where non–linearities can occur in the partial differential equation (PDE), or in subdomains where the right–hand side does not vanish [23, 34]. The same is true for many problems in solid mechanics [31] and in other areas of research. A very straightforward and promising technique for the coupling of FEM and BEM was proposed by M. Costabel [7] and others [6, 33] In the different subdomains of a non–overlapping domain decomposition, we use either the standard finite element (FE) Galerkin method or a mixed–type boundary element (BE) Galerkin method which are weakly coupled over the coupling boundaries (interfaces) Γ_C. The mixed BE Galerkin method makes use of the full Cauchy data representation on the BE subdomain boundaries via the Calderón projector.

The main aim of the project was the design, analysis and implementation of fast and well adapted parallel solvers for large-scale coupled FE/BE–equations approximating plane, linear and nonlinear magnetic field problems including technical magnetic field problems (e.g. electric motors). To be specific, we consider a characteristical cross-section, which lies in the (x, y)-plane of the $\mathbf{R}^3$, of the original electromagnetic device that is to be modelled. Let us assume that $\Omega_0 \subset \mathbf{R}^2$ is a bounded simply connected domain and that homogeneous

*This research has been supported by the German Research Foundation DFG within the Priority Research Programme "Boundary Element Methods" under the grant La 767/1.

Dirichlet boundary conditions are given on $\Gamma_D = \partial\Omega_0$. Formally the nonlinear magnetic field problem can be written as follows [25]

$$
\begin{aligned}
-\mathrm{div}\ (\nu(x, |\nabla u(x)|)\nabla u(x)) &= \mathrm{S}(x) + \frac{\partial \mathrm{H}_{0\mathrm{y}}(x)}{\partial \mathrm{x}} - \frac{\partial \mathrm{H}_{0\mathrm{x}}(x)}{\partial \mathrm{y}}, \quad x \in \Omega && (1)\\
u(x) &= 0, \quad x \in \Gamma_D && (2)\\
|u(x)| &\to 0 \quad \text{for } |x| \to \infty, && (3)
\end{aligned}
$$

with $\Omega := R^2 \setminus \bar{\Omega}_0$. The solution u is the z–component A_z of the vector potential $\vec{A} = (A_x, A_y, A_z)^T$ introduced in the Maxwell equation. The component of the current density, which acts orthogonal to the cross-section being considered, is represented by $\mathrm{S}(x)$, whereas $\mathrm{H}_{0\mathrm{x}}$ and $\mathrm{H}_{0\mathrm{y}}$ stand for sources associated with permanent magnets that may occur and $\nu(.)$ denotes a coefficient depending on the material and on the gradient $|\nabla u(x)|$ (induction). Now, we introduce the exterior domain Ω_+ by defining a, so called, coupling boundary $\Gamma_+ := \partial\Omega_+$. The definition of Γ_+ is restricted by the conditions

$$
\nu(x) = \nu_p \ \ \forall x \in \Omega_+, \ \ (\mathrm{supp}\ \mathrm{S} \cup \mathrm{supp}\ \mathrm{H}_0) \subset \bar{\Omega}_-, \ \ \mathrm{diam}(\bar{\Omega}_0 \cup \bar{\Omega}_-) < 1, \tag{4}
$$

where $\Omega_- := R^2 \setminus (\bar{\Omega}_+ \cup \bar{\Omega}_0)$. Note that the condition $\mathrm{diam}(\bar{\Omega}_0 \cup \bar{\Omega}_-) < 1$ is only technical and can be fulfilled by scaling the problem appropriately. Besides the decomposition $\bar{\Omega} = \bar{\Omega}_- \cup \bar{\Omega}_+$, we allow the inner domain Ω_- to be decomposed further following the natural decomposition of Ω_- according to the change of data:

$$
\bar{\Omega}_- = \bigcup_{j=1}^{N_M} \bar{\hat{\Omega}}_j, \quad \text{with} \quad \hat{\Omega}_i \cap \hat{\Omega}_j = \emptyset \quad \forall i \neq j. \tag{5}
$$

In Section 2 (see also Appendix A), we present an automatic and adaptive domain decomposition procedure providing such a decomposition of Ω into p subdomains (p = number of processors to be used) and such controlling data for the distributed mesh generator [11] that we can expect a well load-balanced performance of our solver.

In Section 3, we consider linear plane magnetic field problems for which a domain decomposition according to Section 2 is available. Now, we can make use of the advantages of a mixed variational DD FE/BE discretization and propose an algorithm for solving the linear coupled FE/BE equations. First of all, the coupled FE/BE equations can be reformulated as a linear system with a symmetric, but indefinite system matrix. We provide a preconditioning and a parallelization of Bramble/Pasciak's Conjugate Gradient (CG) method [2] applied to the symmetric and indefinite system (18). The components of the preconditioner can be chosen such that the resulting algorithm is, at least, almost asymptotically optimal with respect to the operation count and quite robust with respect to complicated geometries, jumping coefficients and mesh grading near singularities (see numerical results given in Sect. 3 and in [30]). Using a special DD data distribution, we parallelize the preconditioning equation and the remaining algorithm in such a way that the same amount of communication is needed as in the earlier introduced and well studied parallel PCG for solving symmetric and positive definite FE equations [17, 18] (see Appendix B).

Section 4 is devoted to the description of the Full-DD-Newton–Solver for nonlinear magnetic field problems. In every nested Newton step we use basically the linear DD–solver given in Section 3.

In Section 5, we apply our linear DD–solver to plane linear elasticity problems modelled by Lamè's system of PDEs. An appropriate adaptation of the components of the DD–preconditioner results in a parallel solver the efficiency of which is comparable to that of the solver for the linear magnetic field problems described in Section 3.

In Section 6, we briefly describe the software package FEM⊗BEM [14] and draw some conclusions. All numerical results presented in this paper were obtained by the use of the package FEM⊗BEM. The code runs on various parallel computers and programming platforms including PVM (see, e.g., [28]).

2 Adaptive Domain Decomposition Preprocessing

2.1 The DD-Data Partitioning

In this section, we focus our interest on how a decomposition of the domain Ω into a given number of subdomains can be obtained from the natural decomposition into domains according to the change of materials (5). We are interested in well load-balanced decompositions especially in the case of discretizations which are adapted to singularities.

We assume that a triangular-based description of the geometry of the problem under consideration is given. Besides the geometrical data each triangle is characterized by a parameter pointing to that of the N_M material-regions the triangle belongs to. Note that interfaces between different materials, i.e. the boundaries of the $\hat{\Omega}_j$'s (cf. (5)), are represented by edges of the triangulation. We are interested in a decomposition into $p \geq N_M$ subdomains

$$\bar{\Omega} = \bigcup_{i \in \mathcal{I}} \bar{\Omega}_i, \quad \text{where} \quad \Omega_p := \Omega_+ \quad \text{and} \quad \bar{\hat{\Omega}}_j = \bigcup_{i \in \mathcal{I}_j} \bar{\Omega}_i \quad \forall j = 1, \ldots, N_M \tag{6}$$

where the sets of indices are given by $\mathcal{I} := \{1, \ldots, p\}$ and

$$\mathcal{I}_j \subset \mathcal{I}^\star := \{1, \ldots, p-1\}, \quad \bigcup_{j=1}^{N_M} \mathcal{I}_j = \mathcal{I}^\star, \quad \mathcal{I}_j \cap \mathcal{I}_k = \emptyset \quad \forall j \neq k,$$

i.e., the subdomains $\hat{\Omega}_j$ determined by the materials may be decomposed further (see, e.g., [19]). We assume that there exist open balls $B_{\underline{r}_i}$ and $B_{\overline{r}_i}$ $(i \in \mathcal{I}^\star)$ with positive radii $\underline{r}_i$ and $\overline{r}_i$, such that $B_{\underline{r}_i} \subset \Omega_i \subset B_{\overline{r}_i}$ and $0 < \underline{c} \leq \overline{r}_i / \underline{r}_i \leq \overline{c}$ $\forall i \in \mathcal{I}^\star$ with fixed (i-independent) constants $\underline{c}$ and $\overline{c}$. Note, in the case of Ω being bounded we had $\mathcal{I}^\star := \{1, \ldots, p\}$ and in the following all terms induced by p, which then stands for the exterior domain, would vanish.

Although the algorithm being used for decomposing Ω is based upon a given triangulation of the domain it is of advantage to use a special DD-data structure as input for DD-based algorithms running on massively parallel computers.

Thus, starting off with a given triangular-based geometrical description ($*.tri$–file) we wish to end up with a well-balanced decomposition of our problem which is described by using some DD-data format ($*.dd$–file). Figure 1 on the right shows the interactions between the preprocessing codes *Decomp, Tri2DD* and *AdapMesh* and the file-types $*.tri$, $*.dd$, and $*.fb$ being involved. In the simplest case the process starts, on the top of the diagram, with applying *Decomp* to a $*.tri$–file which results in a decomposition as defined in (6), i.e. each triangle is assigned to one of the Ω_i's, $i \in \mathcal{I}$. The output of this process is also a $*.tri$–file which then is converted into a $*.dd$–file by applying the program *Tri2DD.* In our case, such a $*.dd$–file is the input for the parallel code FEM⊕BEM [14]. Because of the simple structure of the DD-data format being used it is quite easy to implement additional refinement information concerning, e.g., singularities into the $*.dd$–file. In the latter case the mesh created from this file may differ significantly from the one described by the original $*.tri$–file. As a consequence we have to expect a bad load balance. This problem can be solved by applying the program *AdapMesh* which simulates the mesh generation as it occurs in the parallel program using, optionally, adaptivity information which may be obtained from a coarse-grid computation. Thus, *AdapMesh* creates a $*.tri$–file which is the input for restarting the cycle with *Decomp.*

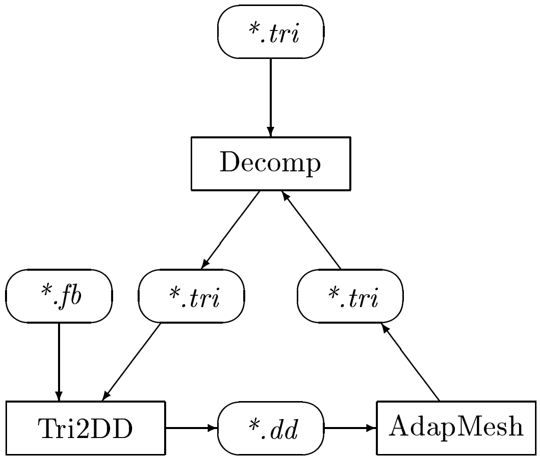

Figure 1: The Preprocessing.

Note, in the optimal case with respect to the load balance the mesh used for computations (i.e. the one created from a $*.dd$–file within FEM⊗BEM) would coincide with the mesh being used for the decomposition.

2.2 A Short Description of the Preprocessing Codes

At this place, we are going to explain the codes and the main ideas they are based on. More information and technical details can be found in the forthcoming documentation [12]. First we give a short description of the codes.

Decomp decomposes single-material domains using the spectral bisection method (sbm) [42]. That is, as long as there are less than p subdomains the largest subdomain according to the number of triangles is divided into two new subdomains by the sbm. As a result, each triangle is assigned to one of the subdomains.

Tri2DD converts triangular-based data into the DD-data format. The algorithm is based on the definition of edges which then define the faces. Note, interfaces between different materials will be maintained as they were given in the original $*.tri$-file. On the other hand the artificially created boundaries within one material are smoothed.

AdapMesh creates a mesh based on DD-data using, optionally, adaptivity information. The resulting $*.tri$-file can be used as input for *Decomp.*

During the preprocessing we are concerned with two types of describing data. The first one ($*.tri$-files) is based on nodes (characterized by their coordinates) which define the edges (straight lines or arcs of circles are allowed) and, finally, triangles defined by their edges. Each of the triangles is characterized by two additional parameters, where only one of them pointing to the material the triangle belongs to has to be initialized from the very beginning. The second one describes the mapping of the triangles on the subdomains and it is defined as a result of decomposing Ω.

The second type of data ($*.dd$-files) follows the DD-data format described in [14]. It is based on defining cross-points numbered globally which then define edges. Main objects are faces described by edges. The faces, or a union of them, are mapped onto the array of processors. The $*.fb$-files are auxiliary files (optional) and contain controlling data or fixed cross-points as it occurs in our example (see below).

2.3 Example: Direct Current Motor

Now we apply our preprocessing tools to a model of a direct current motor (dc-motor) with permanent excitation (see also Fig. 2 and 3 in [24] for a detailed description of the machine). We start with a $*.tri$-file with 18 different material regions as shown on the left in Figure 5 (Appendix A) . After applying *Decomp, Tri2DD* and one full cycle we have obtained the $*.dd$-file which represents the decomposition as shown on the right in Figure 5. Note the additional cross-points especially on the outer circle which have been pre-defined in a $*.fb$–file. In our case no additional refinement information have been used so that the second decomposition (Figure 5) did not differ significantly from the first one which were already obtained after applying *Decomp* and *Tri2DD.* Figure 6 (Appendix A) shows, on the left, the mesh being finally used as initial mesh for computations and, on the right, the equipotential lines of the solution.

3 Parallel Solution of Linear Coupled BE/FE–Equations Approximating Linear Magnetic Field Problems

3.1 A Mixed Variational Formulation

Let us first consider a linear ($\nu = \nu(x)$) magnetic field problem of the form (1) - (3) for which a domain decomposition according to Section 2 is available. In particular, we assume that the index set $\mathcal{I} = \mathcal{I}_F \cup \mathcal{I}_B$ can be decomposed into two disjoint sets of indices $\mathcal{I}_F$ and $\mathcal{I}_B$ such that

$$p \in \mathcal{I}_B, \tag{7}$$

$$(\text{supp S}(.) \cup \text{supp H}_0(.)) \cap \Omega_i = \emptyset \quad \forall i \in \mathcal{I}_B, \tag{8}$$

$$\nu(x) = \nu_i = const \quad \forall i \in \mathcal{I}_B. \tag{9}$$

For each Ω_i $(i \in \mathcal{I})$ the index i belongs to one of the two index sets $\mathcal{I}_B$ and $\mathcal{I}_F$ according to the discretization method applied to Ω_i, where $\mathcal{I}_B$ and $\mathcal{I}_F$ stand for BEM and FEM, respectively.

Following M.Costabel [7], G.C.Hsiao and W.L.Wendland [33] and others, we can rewrite the weak formulation of the boundary value problem (1) - (3) by means of partial integration in the boundary element subdomains Ω_i, $i \in \mathcal{I}_B$ and by the use of Calderón's representation of the full Cauchy data as a mixed DD coupled domain and boundary integral variational problem: Find $(\lambda, u) \in \mathbf{V} := \mathbf{\Lambda} \times \mathbf{U}_0$ such that

$$a(\lambda, u; \eta, v) = \langle F, v \rangle \quad \forall (\eta, v) \in \mathbf{V}, \tag{10}$$

where

$$\begin{aligned}
a(\lambda, u; \eta, v) &:= a_B(\lambda, u; \eta, v) + a_F(u, v) \\
a_B(\lambda, u; \eta, v) &:= \sum_{i \in \mathcal{I}_B \setminus \{p\}} \nu_i \Big\{ \langle \mathcal{D}_i u_i, v_i \rangle_{\Gamma_i} + \frac{1}{2} \langle \lambda_i, v_i \rangle_{\Gamma_i} + \langle \lambda_i, \mathcal{K}_i v_i \rangle_{\Gamma_i} + \\
&\qquad \langle \eta_i, \mathcal{V}_i \lambda_i \rangle_{\Gamma_i} - \langle \eta_i, \mathcal{K}_i u_i \rangle_{\Gamma_i} - \frac{1}{2} \langle \eta_i, u_i \rangle_{\Gamma_i} \Big\} \\
&\qquad + \nu_p \Big\{ \langle \mathcal{D}_p u_p, v_p \rangle_{\Gamma_+} - \frac{1}{2} \langle \lambda_p, v_p \rangle_{\Gamma_+} + \langle \lambda_p, \mathcal{K}_p v_p \rangle_{\Gamma_+} + \\
&\qquad \langle \eta_p, \mathcal{V}_p \lambda_p \rangle_{\Gamma_+} - \langle \eta_p, \mathcal{K}_p u_p \rangle_{\Gamma_+} + \frac{1}{2} \langle \eta_p, u_p \rangle_{\Gamma_+} \Big\} \\
a_F(u, v) &:= \sum_{i \in \mathcal{I}_F} \int_{\Omega_i} \nu(x) \nabla^T u(x) \nabla v(x)\, dx \\
\langle F, v \rangle &:= \sum_{i \in \mathcal{I}_F} \int_{\Omega_i} \left[\mathrm{S}(x) v(x) - \mathrm{H}_{0\mathrm{y}}(x) \frac{\partial v(x)}{\partial \mathrm{x}} + \mathrm{H}_{0\mathrm{x}}(x) \frac{\partial v(x)}{\partial \mathrm{y}} \right] dx \\
\langle \lambda_i, v_i \rangle_{\Gamma_i} &:= \int_{\Gamma_i} \lambda_i v_i \, ds \text{ and } v_i = v|_{\partial \Omega_i},\ u_i = u|_{\partial \Omega_i},\ \Gamma_i := \partial \Omega_i,
\end{aligned}$$

with the well-known boundary integral operators $\mathcal{V}_i, \mathcal{K}_i, \mathcal{D}_i$ defined by the relation

$$\begin{aligned}
\mathcal{V}_i \lambda_i(x) &:= \int_{\Gamma_i} \mathcal{E}(x, y) \lambda_i(y)\, ds_y \\
\mathcal{K}_i v_i(x) &:= \int_{\Gamma_i} \partial_y \mathcal{E}(x, y) v_i(y)\, ds_y \\
\mathcal{D}_i u_i(x) &:= -\partial_x \int_{\Gamma_i} \partial_y \mathcal{E}(x, y) u_i(y)\, ds_y
\end{aligned} \tag{11}$$

and with the fundamental solution

$$\mathcal{E}(x, y) = -\frac{1}{2\pi} \log |x - y| \tag{12}$$

of the Laplacian. The mapping properties of the boundary integral operators (11) on Sobolev spaces are now well known [8]. The spaces $\mathbf{U}_0$ and $\mathbf{\Lambda}$ are defined by the relations

$$\begin{array}{rcl}\mathbf{U}_0 & := & \{u \in \mathrm{H}^1(\Omega_-) : \ u|_{\Gamma_{BE}} \in \mathrm{H}^{1/2}(\Gamma_{BE}), u|_{\partial\Omega_0} = 0\} \\ \mathbf{\Lambda} & := & \left\{\lambda = (\lambda_i)_{i\in\mathcal{I}_B} : \ \lambda_i \in \mathrm{H}^{-1/2}(\Gamma_i), \ i \in \mathcal{I}_B\right\} = \prod_{i\in\mathcal{I}_B} \Lambda_i,\end{array} \tag{13}$$

with $\Lambda_i = \mathrm{H}^{-1/2}(\Gamma_i), \ i \in \mathcal{I}_B$. Further we use the notation $\Gamma_{BE} := \bigcup_{i\in\mathcal{I}_B} \partial\Omega_i \setminus \Gamma_D, \Gamma_{FE} := \bigcup_{i\in\mathcal{I}_F} \partial\Omega_i \setminus \Gamma_D, \Gamma_C := \Gamma_{BE} \cup \Gamma_{FE}$ and $\Omega_F := \bigcup_{i\in\mathcal{I}_F} \Omega_i$. Introducing in $\mathbf{V} := \mathbf{\Lambda} \times \mathbf{U}_0$ the norm

$$\|(\lambda, u)\|_{\mathbf{V}} := \left(\|\lambda\|^2_{\Lambda} + \|u|_{\Gamma_{BE}}\|^2_{\mathrm{H}^{1/2}(\Gamma_{BE})} + \|u\|^2_{\mathrm{H}^1(\Omega_F)}\right)^{1/2} \tag{14}$$

with

$$\|\lambda\|^2_{\Lambda} = \sum_{i\in\mathcal{I}_B} \|\lambda_i\|^2_{\mathrm{H}^{-1/2}(\Gamma_i)} \quad \text{and} \quad \|u|_{\Gamma_{BE}}\|^2_{\mathrm{H}^{1/2}(\Gamma_{BE})} = \sum_{i\in\mathcal{I}_B} \|u_i\|^2_{\mathrm{H}^{1/2}(\Gamma_i)},$$

then one can prove that the bilinear form a(.,.) is $\mathbf{V}$–elliptic and $\mathbf{V}$–bounded provided that the domain decomposition satisfies the conditions imposed on above (see also [33]). Therefore, the existence and uniqueness of the solution are a direct consequence of the Lax–Milgram theorem.

3.2 The Coupled BE/FE Discretization

Now, we can define the nodal FE/BE basis of piecewise linear trial functions based upon a regular triangulation of the subdomains $\Omega_i, i \in \mathcal{I}_F$ and the according discretization of the boundary pieces $\Gamma_{ij} = \bar{\Omega}_i \cap \bar{\Omega}_j, \ i,j \in \mathcal{I}$:

$$\Phi = [\phi_1, \ldots, \phi_{N_\Lambda}, \phi_{N_\Lambda+1}, \ldots, \phi_{N_\Lambda+N_C}, \phi_{N_\Lambda+N_C+1}, \ldots, \phi_N],$$

where $N = N_\Lambda + N_C + N_I$ and $N_I = \sum_{i\in\mathcal{I}_F} N_{I,i}, \ N_\Lambda = \sum_{i\in\mathcal{I}_B} N_{\Lambda,i}$. Here, $\phi_1, \ldots, \phi_{N_\Lambda}$ are the basis functions for approximating λ on $\Gamma_i, \ i \in \mathcal{I}_B, \ \phi_{N_\Lambda+1}, \ldots, \ \phi_{N_\Lambda+N_C}$ represent u on Γ_C and $\phi_{N_\Lambda+N_C+1}, \ldots, \phi_{N_\Lambda+N_C+N_I}$ approximate u in $\Omega_i, \ i \in \mathcal{I}_F$. The definition of the finite dimensional subspaces of $\mathbf{\Lambda}, \mathbf{U}_0$ and $\mathbf{V}$

$$\begin{array}{rcl}\mathbf{\Lambda}_h & := & \mathrm{span}\ [\phi_1, \phi_2, \ldots, \phi_{N_\Lambda}], \\ \mathbf{U}_h & := & \mathrm{span}\ [\phi_{N_\Lambda+1}, \ldots, \phi_{N_\Lambda+N_C}, \phi_{N_\Lambda+N_C+1}, \ldots, \phi_N], \\ \mathbf{V}_h & := & \mathbf{\Lambda}_h \times \mathbf{U}_h\end{array}$$

allows us to formulate the discrete problem as follows: Find $u_h \in \mathbf{V}_h$ such that

$$a(u_h, v_h) = \langle F, v_h\rangle \quad \forall v_h \in \mathbf{V}_h. \tag{15}$$

The isomorphism $\Phi : \mathbf{R}^N \to \mathbf{V}_h$ leads to the linear system:

$$\begin{pmatrix} K_\Lambda & -K_{\Lambda C} & 0 \\ K_{C\Lambda} & K_C & K_{CI} \\ 0 & K_{IC} & K_I \end{pmatrix} \begin{pmatrix} \mathbf{u}_\Lambda \\ \mathbf{u}_C \\ \mathbf{u}_I \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\Lambda \\ \mathbf{f}_C \\ \mathbf{f}_I \end{pmatrix}, \tag{16}$$

where the block entries are defined by

$$\begin{array}{rcl}(K_\Lambda \mathbf{u}_\Lambda, \mathbf{v}_\Lambda) & = & \displaystyle\sum_{i\in\mathcal{I}_B} \nu_i \langle \eta_i, \mathcal{V}_i \lambda_i\rangle_{\Gamma_i} \quad \text{with } \lambda_i = \Phi_{\Lambda_i}\mathbf{u}_{\Lambda_i}, \eta_i = \Phi_{\Lambda_i}\mathbf{v}_{\Lambda_i}, \\ (K_{C\Lambda}\mathbf{u}_\Lambda, \mathbf{v}_C) & = & \displaystyle\sum_{i\in\mathcal{I}_B\setminus\{p\}} \nu_i\{\langle\lambda_i, \mathcal{K}_i v_i\rangle_{\Gamma_i} + \frac{1}{2}\langle\lambda_i, v_i\rangle_{\Gamma_i}\} \\ & & \displaystyle +\nu_p\{\langle\lambda_p, \mathcal{K}_p v_p\rangle_{\Gamma_p} - \frac{1}{2}\langle\lambda_p, v_p\rangle_{\Gamma_p}\} \\ K_{\Lambda C} & = & K^T_{C\Lambda} \\ K_C & = & K_{CB} + K_{CF}, \ \text{ with} \\ (K_{CB}\mathbf{u}_C, \mathbf{v}_C) & = & \displaystyle\sum_{i\in\mathcal{I}_B} \nu_i\langle\mathcal{D}_i u_i, v_i\rangle_{\Gamma_i}, \ u_i = \Phi_{C_i}\mathbf{u}_{C_i}, \ v_i = \Phi_{C_i}\mathbf{v}_{C_i} \ \text{ and}\end{array}$$

$$\left(\begin{pmatrix} K_{CF} & K_{CI} \\ K_{IC} & K_I \end{pmatrix} \begin{pmatrix} \mathbf{u}_C \\ \mathbf{u}_I \end{pmatrix}, \begin{pmatrix} \mathbf{v}_C \\ \mathbf{v}_I \end{pmatrix} \right) = \sum_{i \in \mathcal{I}_F} \int_{\Omega_i} \nu(x) \nabla^T u \, \nabla v \, dx,$$

where $u|_{\Omega_F} = \Phi_F \mathbf{u}_F$, $v|_{\Omega_F} = \Phi_F \mathbf{v}_F$. Here, $\Phi_{\Lambda_i} \quad (i \in \mathcal{I}_B)$ and $\Phi_{C_i} \quad (i \in \mathcal{I})$ contain the basis functions for approximating λ and u on $\partial\Omega_i$, respectively. The basis functions in Φ_F are used to approximate u in $\Omega_i \quad (i \in \mathcal{I}_F)$. The FE entries, especially K_I, are sparse matrices, whereas the BE blocks are fully populated.

3.3 The Parallel Solver

3.3.1 Bramble-Pasciak's Transformation and Spectral Equivalence Results

The nonsymmetric, positive definite system (16) can be solved approximately by Bramble/Pasciak's CG method [2]. The method requires a preconditioner C_Λ which can be inverted easily and which fulfills the spectral equivalence inequalities

$$\underline{\gamma}_\Lambda C_\Lambda \le K_\Lambda \le \overline{\gamma}_\Lambda C_\Lambda, \quad \text{with} \quad \underline{\gamma}_\Lambda > 1. \tag{17}$$

With the definitions

$$\begin{aligned} K_1 &= K_\Lambda, \quad \mathbf{f}_1 = \mathbf{f}_\Lambda, \quad K_{12} = K_{21}^T = (-K_{\Lambda C} \quad 0) \\ K_2 &= \begin{pmatrix} K_C & K_{CI} \\ K_{IC} & K_I \end{pmatrix}, \quad \mathbf{f}_2 = \begin{pmatrix} -\mathbf{f}_C \\ -\mathbf{f}_I \end{pmatrix}. \end{aligned}$$

we can reformulate (16) as a symmetric but indefinite system:

$$\begin{pmatrix} K_1 & K_{12} \\ K_{21} & -K_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}. \tag{18}$$

Following Bramble and Pasciak [2] this system can be transformed into

$$G\,\mathbf{u} = \mathbf{p}, \text{ where} \tag{19}$$

$$G := \begin{pmatrix} C_\Lambda^{-1} K_1 & C_\Lambda^{-1} K_{12} \\ K_{21} C_\Lambda^{-1} (K_1 - C_\Lambda) & K_2 + K_{21} C_\Lambda^{-1} K_{12} \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} C_\Lambda^{-1} \mathbf{f}_1 \\ K_{21} C_\Lambda^{-1} \mathbf{f}_1 - \mathbf{f}_2 \end{pmatrix}.$$

Then, the matrix G is self-adjoint and positive definite with respect to the scalar product $[.,.]$ which is defined by

$$[\mathbf{w}, \mathbf{v}] := ((K_1 - C_\Lambda)\,\mathbf{w}_1, \mathbf{v}_1) + (\mathbf{w}_2, \mathbf{v}_2). \tag{20}$$

Moreover, G is spectrally equivalent to the regularisator R, where

$$R := \begin{pmatrix} I & 0 \\ 0 & K_2 + K_{21} K_1^{-1} K_{12} \end{pmatrix}.$$

Bramble and Pasciak [2] proved the spectral equivalence inequalities

$$\underline{\lambda}\,[R\,\mathbf{v}, \mathbf{v}] \le [G\,\mathbf{v}, \mathbf{v}] \le \overline{\lambda}\,[R\,\mathbf{v}, \mathbf{v}] \quad \forall \mathbf{v} \in \mathbf{R}^N, \tag{21}$$

where

$$\underline{\lambda} = \left(1 + \frac{\alpha}{2} + \sqrt{\alpha + \frac{\alpha^2}{4}} \right)^{-1} \quad \text{and} \quad \overline{\lambda} = \frac{1 + \sqrt{\alpha}}{1 - \alpha} \tag{22}$$

with $\alpha = 1 - (1/\overline{\gamma}_\Lambda)$. Thus, we have to find a preconditioner C_2 for the matrix

$$K_2 + K_{21} K_1^{-1} K_{12} = \begin{pmatrix} K_C + K_{C\Lambda} K_\Lambda^{-1} K_{\Lambda C} & K_{CI} \\ K_{IC} & K_I \end{pmatrix}. \tag{23}$$

The DD preconditioner defined by

$$C_2 = \begin{pmatrix} I_C & K_{CI}B_I^{-T} \\ 0 & I_I \end{pmatrix} \begin{pmatrix} C_C & 0 \\ 0 & C_I \end{pmatrix} \begin{pmatrix} I_C & 0 \\ B_I^{-1}K_{IC} & I_I \end{pmatrix} \tag{24}$$

is spectrally equivalent to $K_2 + K_{21}K_1^{-1}K_{12}$ if we have preconditioners C_I and C_C fulfilling the inequalities

$$\underline{\gamma}_C C_C \leq \tilde{S}_C + K_{C\Lambda}K_\Lambda^{-1}K_{\Lambda C} \leq \overline{\gamma}_C C_C, \tag{25}$$

$$\underline{\gamma}_I C_I \leq K_I \leq \overline{\gamma}_I C_I, \tag{26}$$

where $\tilde{S}_C = K_C - K_{CI}K_I^{-1}K_{IC} + K_{CI}(K_I^{-1} - B_I^{-T})K_I(K_I^{-1} - B_I^{-1})K_{IC}$, and B_I is an appropriately chosen non-singular matrix [37].

Lemma 1 *If the symmetric and positive definite block preconditioners $C_I = diag(C_{I,i})_{i\in\mathcal{I}_F}$ and C_C satisfy the spectral equivalence inequalities (25) and (26) with positive constants $\underline{\gamma}_C$, $\overline{\gamma}_C$, $\underline{\gamma}_I$, $\overline{\gamma}_I$, then the spectral equivalence inequalities*

$$\underline{\gamma}_2 C_2 \leq K_2 + K_{21}K_1^{-1}K_{12} \leq \overline{\gamma}_2 C_2 \tag{27}$$

hold for the preconditioner C_2 defined in (24) with the constants

$$\underline{\gamma}_2 = \min\{\underline{\gamma}_C, \underline{\gamma}_I\} \left(1 - \sqrt{\tfrac{\mu}{1+\mu}}\right), \quad \overline{\gamma}_2 = \max\{\overline{\gamma}_C, \overline{\gamma}_I\} \left(1 + \sqrt{\tfrac{\mu}{1+\mu}}\right). \tag{28}$$

Here $\mu = \rho(S_C^{-1}T_C)$ denotes the spectral radius of $S_C^{-1}T_C$, with the FE Schur complement S_C and the operator T_C being defined by

$$S_C = \overset{o}{K}_C - \overset{o}{K}_{CI}K_I^{-1}\overset{o}{K}_{IC} \quad \textit{and} \quad T_C = \overset{o}{K}_{CI}(K_I^{-1} - B_I^{-T})K_I(K_I^{-1} - B_I^{-1})\overset{o}{K}_{IC},$$

respectively. $\overset{o}{K}_C$, $\overset{o}{K}_{CI}$ and $\overset{o}{K}_{IC}$ denote the non-zero FE blocks of K_C, K_{CI} and K_{IC}, respectively.

The proof is given in [37], it applies the classical FE DD spectral equivalence result proved in [17, 18]. With (22), we conclude the following theorem.

THEOREM 1 *If the conditions imposed on C_Λ, C_C, C_I, and B_I, especially (17), (25) and (26) are satisfied, then the FE/BE DD preconditioner*

$$C = diag(I_1, C_2) \tag{29}$$

is self-adjoint and positive definite with respect to the inner product $[.,.]$ and satisfies the spectral equivalence inequalities

$$\underline{\gamma}\,[C\,\mathbf{v},\mathbf{v}] \leq [G\,\mathbf{v},\mathbf{v}] \leq \overline{\gamma}\,[C\,\mathbf{v},\mathbf{v}] \quad \forall \mathbf{v}\in\mathbf{R}^N, \tag{30}$$

with the constants

$$\underline{\gamma} = \underline{\lambda}\min\left\{1, \underline{\gamma}_2\right\} \quad \textit{and} \quad \overline{\gamma} = \overline{\lambda}\max\left\{1, \overline{\gamma}_2\right\},$$

where $\underline{\lambda}$, $\overline{\lambda}$, $\underline{\gamma}_2$, $\overline{\gamma}_2$ are given in (22) and (28), respectively.

3.3.2 The Parallel PCG Algorithm

For the vectors belonging to the inner coupling boundary Γ_C we define two types of distribution called overlapping (type 1) and adding (type 2):

<u>type 1:</u> $\mathbf{u}_C, \mathbf{w}_C, \mathbf{s}_C$ are stored in P_i as $\mathbf{u}_{C,i} = A_{C,i}\mathbf{u}_C$ (analogous $\mathbf{w}_{C,i}$, $\mathbf{s}_{C,i}$)
<u>type 2:</u> $\mathbf{r}_C, \mathbf{v}_C, \mathbf{f}_C$ are stored in P_i as $\mathbf{r}_{C,i}, \mathbf{v}_{C,i}, \mathbf{f}_{C,i}$ such that $\mathbf{r}_C = \sum_{i=1}^{p} A_{C,i}^T \mathbf{r}_{C,i}$ (analogous $\mathbf{v}_C, \mathbf{f}_C$),

where the matrices $A_{C,i}$ are the "C-block" of the Boolean subdomain connectivity matrix A_i which maps some overall vector of nodal parameters into the superelement vector of parameters associated with the subdomain $\bar{\Omega}_i$ only. P_i denotes the i^{th} processor.

Using this notation and the operators introduced in the previous section we can formulate an improved version of the PCG-algorithm presented in [37] with a given accuracy ε as stopping criterion. This parallel PCG-algorithm is given in Appendix B.

Note the vectors $\mathbf{z}_i = (\mathbf{z}_{\Lambda,i}, \mathbf{z}_{C,i}, \mathbf{z}_{\Lambda C,i})^T$ and $\mathbf{h}_i = (\mathbf{h}_{\Lambda,i}, \mathbf{h}_{C,i}, \mathbf{h}_{\Lambda C,i})^T$ which have been inserted additionally in order to achieve a synchronization between the FEM and BEM processors especially in step 1 (matrix-times-vector operation). Without this synchronization one has to expect a computation time per iteration which is, depending on the problem, up to 30 per cent higher. The definition of the vector $\mathbf{p}_i$ avoids the computation of $C_{\Lambda,i}\mathbf{r}_{\Lambda,i}$ (occurred originally in step 4) which is not necessarily available ($C_{\Lambda,i}$ is defined such that the inverse operation $C_{\Lambda,i}^{-1}\mathbf{w}_{\Lambda,i}$ can be performed easily).

3.3.3 On the Components of the Preconditioner

The performance of our algorithm depends heavily on the right choice of the components C_Λ, C_C, C_I and B_I defining the preconditioner C (see Theorem 1). C_Λ, C_I and B_I are block-diagonal matrices with the blocks $C_{\Lambda,i}$, $C_{I,i}$ and $B_{I,i}$, respectively. In our experiments, the following components have turned out to be the most efficient ones:

$C_{I,i}$: (**Vmn**) Multigrid V-cycle with m pre- and n post-smoothing steps in the Multiplicative Schwarz Method [18, 15].

$C_{\Lambda,i}$: (**Circ**) Scaled single layer potential BE matrix for a uniformly discretized circle. This matrix is circulant and easily invertible [39].
(**Hyp**) $C_{\Lambda,i}^{-1} = T_i^T \tilde{M}_{h,i}^{-1} K_{C,i} \tilde{M}_{h,i}^{-1} T_i$ as proposed by Steinbach [43].

$B_{I,i}$: (**HExt**) Implicitly defined by hierarchical extension (formally $E_{IC,i} = -B_{I,i}^{-1} K_{IC,i}$) [20].

C_C: (**S-BPX**) Bramble/Pasciak/Xu type preconditioner [44].
(**BPS-D**) Bramble/Pasciak/Schatz type preconditioner [3, 9].
(**mgD**) $K_{C^*}(I_C - M_C)^{-1}$, $(K_{C^*}\mathbf{u}_C, \mathbf{v}_C) = \sum_{i\in\mathcal{I}} \nu_i \langle \mathcal{D}_i u_i, v_i\rangle$, as described in [5].

These preconditioners C_I, C_C, C_Λ, and the basis transformation B_I satisfy the conditions stated in Theorem 1. In particular, inequalities (26) for C_I are fulfilled with constants $\underline{\gamma}_I$, $\overline{\gamma}_I$ independent of the discretization parameter h [18, 15, 20]. The preconditioner C_Λ is scaled such that $\underline{\gamma}_\Lambda > 1$, and $\overline{\gamma}_\Lambda$ in (17) remains independent of h for both, (Circ) and (Hyp). In the case (Hyp), $C_{\Lambda,i}^{-1}$ involves a basis transformation T_i, a modified mass-matrix $\tilde{M}_{h,i}$ and the hypersingular operator $K_{C,i}$, and property (17) for C_Λ is due to properties of the corresponding continuous operators $\mathcal{V}_i^{-1}$ and $\mathcal{D}_i$ [43].

With respect to B_I, the constant μ in (28) can be estimated by

$$\mu \le \eta^{2k}(1 + c_1\, l)^2 \le \eta^{2k}(1 + c_2(\ln h^{-1}))^2,$$

cf. [20], with k being the number of local multigrid iterations, l being the number of grids, the h-independent multigrid rate $\eta < 1$, and the h-independent constants c_i. Thus, μ is independent of h if $k = \mathcal{O}(\ln \ln h^{-1})$.

In the (S-BPX) case, the inequalities in (25) hold with an h-independent constant $\underline{\gamma}_C$, and $\overline{\gamma}_C \le c_3(1+\mu)$ [20, 44]. Therefore, we can prove for (S-BPX) that $\overline{\gamma}/\underline{\gamma} \le c_4(1+\mu)(\sqrt{\mu}+\sqrt{1+\mu})^2 = \mathcal{O}(1)$ if $k = \mathcal{O}(\ln \ln h^{-1})$. However, in the range of practical applications, this means $k = 1$! For (BPS-D), the estimate $\overline{\gamma}_C/\underline{\gamma}_C \le c_5(1 + (\ln h^{-1})^2)(1+\mu)$ has been proved [3]. In the case (mgD), C_C arises from the hypersingular operator and C_C^{-1} is realized via a standard multi-grid procedure applied to the global operator (assembled over the subdomains) K_{C^*} which is the discretization of a pseudo-differential operator of order one. K_{C^*} becomes positive definite after implementing the Dirichlet boundary conditions. We can get an estimate of the same type as for (S-BPX). Note that the FE/BE–Schur-complement energy is equivalent to the $\|.\|^2_{\mathrm{H}^{1/2}(\Gamma_C)}$–norm (see, e.g., [3, 5, 9]).

Consequently, we can estimate the numerical effort Q to obtain a relative accuracy ε by $Q = \mathcal{O}(h^{-2} \ln h^{-1} \ln \ln h^{-1} \ln \varepsilon^{-1})$ for the (BPS-D) case, and by $Q = \mathcal{O}(h^{-2} \ln \ln h^{-1} \ln \varepsilon^{-1})$ in the (S-BPX) case, i.e. almost optimal. If a BPX-type extension [38] is applied instead of (HExt) in a nested iteration approach [21], we can prove that $Q = \mathcal{O}(h^{-2})$, i.e., we obtain an optimal method.

Preconditioners $C_{\Lambda,i}$ for $K_{\Lambda,i}$ and C_C for the FE/BE Schur complement derived on the basis of boundary element techniques can also be found in [35, 39]. The construction of efficient FE Schur complement preconditioners was one of the main topics in the research on FE-DD-methods (see Proceedings of the annual DD-conferences since 1987).

3.4 Numerical Results

The electromagnet as shown in Figure 2 serves now as test example for exterior magnetic field problems which lead to the variational form (10). The copper domains (I, II), where we have a

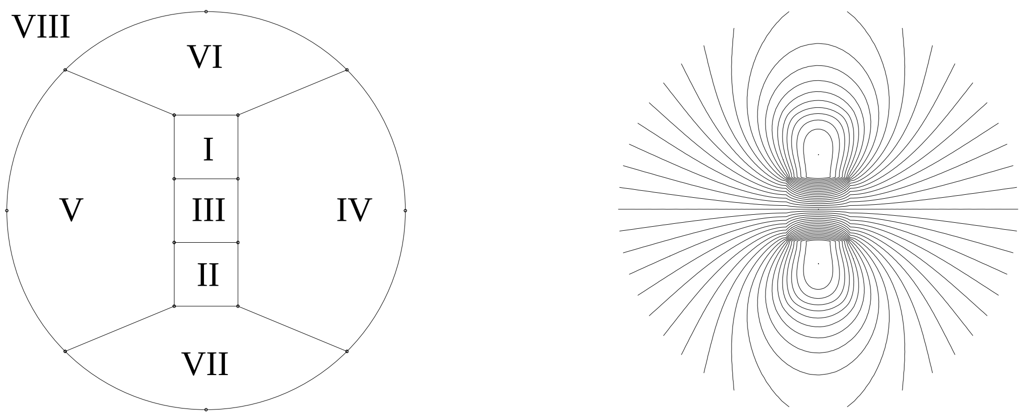

Figure 2: The magnet and the subdomains being used (left) and the equipotential lines of the solution (right).

current density of the strength S and $-$S, respectively, and the iron domain (III), are squares with the edges being 16cm long. The material dependent coefficients (air: IV–VIII) are given by $\nu_{Cu} = 795779.0\,\mathrm{AmV^{-1}s^{-1}}$, $\nu_{air} = 795774.4\,\mathrm{AmV^{-1}s^{-1}}$, $\nu_{Fe} = 1000.0\,\mathrm{AmV^{-1}s^{-1}}$.

We will compare two different coupling procedures. On the one hand we use the natural boundary (of the metallic material) as coupling boundary and on the other hand we introduce a circle with the radius 50cm as coupling boundary. The advantage of the second method is that we obtain circulant matrices which can be generated very fast whereas the first method requires the generation of fully populated matrices. A disadvantage of the second method is

	FEM: I-VII BEM: VIII		FEM: I-III BEM: IV-VIII		FEM: I-III BEM: exterior	
l	I(ε)	CPU	I(ε)	CPU	I(ε)	CPU
1	15	0.6	17	0.5	14	0.3
2	19	0.9	18	1.0	19	0.6
3	20	1.6	19	1.9	20	1.7
4	21	5.2	21	5.2	21	5.9
5	23	20.5	22	18.8	22	22.5
N(5)	67329		18429		16129	

Table 1: Number of unknowns (N), iteration count ($I(\varepsilon)$, $\varepsilon = 10^{-6}$), CPU time in seconds. The experiments were carried out on a Power-XPlorer using 8 or 4 processors, respectively.

that additional subdomains and, thus, additional unknowns have to be introduced. For this example, the uniqueness of the solution is guaranteed by the radiation condition, even if no Dirichlet boundary Γ_D is present. The radiation condition is implicitly contained in our BE discretization.

Numerical results are given in Table 1. The operators C_Λ, C_C, C_I, B_I have been chosen as follows:

$$\begin{array}{llll} C_{\Lambda,i} & : \text{Circ } (i=8) \text{ and Hyp } (i \in \mathcal{I}_B \setminus \{8\}) & C_C & : \text{S-BPX} \\ C_{I,i} & : \text{V11 } (i \in \mathcal{I}_F) & B_{I,i} & : \text{HExt } (i \in \mathcal{I}_F). \end{array}$$

Comparing the three choices of the subdomains and their discretization, we observe that in the first choice (column 1) much time is spent for handling the FEM subdomains IV, V with many interior nodes. We may conclude that the BEM (column 2) is recommended for subdomains with a high ratio between the numbers of interior FEM nodes and the boundary nodes. The choice of the rectangular coupling boundary (column 3) increases the BEM system generation effort, but the total time remains nearly the same since the number of BEM unknowns (and subdomains) is reduced. Finally we observe that the number of iterations is independent of the combination of BE/FE discretizations being used.

4 Parallel Solution of Coupled BE/FE–Equations Approximating Nonlinear Magnetic Field Problems

4.1 The Nested-DD-Newton-Solver

Let us consider now a nonlinear magnetic field problem, i.e., we allow ferromagnetic materials with non-constant permeability ν to be in the FE subdomains. Then, the mixed DD coupled variational problem can be written as follows: Find $(\lambda, u) \in \mathbf{V} := \mathbf{\Lambda} \times \mathbf{U}_0$ such that

$$a_N(\lambda, u; \eta, v) = \langle F, v \rangle \quad \forall (\eta, v) \in \mathbf{V}, \tag{31}$$

where

$$\begin{aligned} a(\lambda, u; \eta, v) &:= a_B(\lambda, u; \eta, v) + a_{FN}(u, v) \\ a_{FN}(u, v) &:= \sum_{i \in \mathcal{I}_F} \int_{\Omega_i} \nu(x, |\nabla u|) \nabla^T u(x) \nabla v(x)\, dx \end{aligned}$$

and a_B being defined as in (10). We refer to [25, 30] for the analysis of nonlinear magnetic field problems. Consequently, the discretization results in a nonlinear system [26]

$$K \begin{pmatrix} \mathbf{u}_\Lambda \\ \mathbf{u}_C \\ \mathbf{u}_I \end{pmatrix} = K_F \begin{pmatrix} \mathbf{u}_C \\ \mathbf{u}_I \end{pmatrix} + K_B \cdot \begin{pmatrix} \mathbf{u}_\Lambda \\ \mathbf{u}_C \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\Lambda \\ \mathbf{f}_C \\ \mathbf{f}_I \end{pmatrix}, \tag{32}$$

where the nonlinear operator $K : \mathbf{R}^N \longrightarrow \mathbf{R}^N$ can be split up into the nonlinear operator $K_F : \mathbf{R}^{N_C+N_I} \longrightarrow \mathbf{R}^N$ originating from the nonlinear form a_{FN} and the linear operator $K_B : \mathbf{R}^{N_\Lambda+N_C} \longrightarrow \mathbf{R}^N$ originating from a_B [26].

The nonlinear system (32) is solved by Newton's method, see [26]. In this algorithm, the linear Newton defect system (35) with the Jacobi matrix $K'[\mathbf{u}]$ can be written as

$$K'\left[\begin{pmatrix}\mathbf{u}_\Lambda\\ \mathbf{u}_C\\ \mathbf{u}_I\end{pmatrix}\right]\cdot\begin{pmatrix}\mathbf{w}_\Lambda\\ \mathbf{w}_C\\ \mathbf{w}_I\end{pmatrix} = K_F'\left[\begin{pmatrix}\mathbf{u}_C\\ \mathbf{u}_I\end{pmatrix}\right]\cdot\begin{pmatrix}\mathbf{w}_C\\ \mathbf{w}_I\end{pmatrix} + K_B\cdot\begin{pmatrix}\mathbf{w}_\Lambda\\ \mathbf{w}_C\end{pmatrix} = \begin{pmatrix}\mathbf{d}_\Lambda\\ \mathbf{d}_C\\ \mathbf{d}_I\end{pmatrix}. \tag{33}$$

It can be rewritten in a block form similar to (16),

$$\begin{pmatrix} K_\Lambda & -K_{\Lambda C} & 0\\ K_{C\Lambda} & J_C & J_{CI}\\ 0 & J_{IC} & J_I\end{pmatrix}\begin{pmatrix}\mathbf{w}_\Lambda\\ \mathbf{w}_C\\ \mathbf{w}_I\end{pmatrix} = \begin{pmatrix}\mathbf{d}_\Lambda\\ \mathbf{d}_C\\ \mathbf{d}_I\end{pmatrix}, \tag{34}$$

and can be solved by a PCG method as described in Subsection 3.3. Further, we apply the "nested iteration" method [21, 24], i.e. we generate a multilevel sequence of coupled FE/BE discretizations denoted by the grid numbers $q = 1, \dots, l$. We begin with solving the nonlinear system by Newton's method on the coarsest grid $q = 1$. Then we take the approximate solution on the grid $q-1$ interpolated onto the finer grid q as an initial approximation for Newton's method on the grid q, for $q = 2, \dots, l$. This allows us to "catch" the nonlinearity on the coarsest grid, see [24, 26, 28, 30].

The result of the parallel algorithm presented in the following will be the approximate solution $\mathbf{u}_l^\star$ on the fine grid $q = l$ with relative accuracy ε (nested iteration ε).

Algorithm PNN (Parallel Nested Newton)

Step 0

Initialization of the grid number:

(0.)(P) $q := 1$.

Step 1

Set the initial solution for grid q:

(1.1)(P) `IF` $q = 1$ `THEN` $\mathbf{u}_q^0 = \mathbf{0}$;

(1.2)(P) `IF` $q > 1$ `THEN` $\mathbf{u}_q^0 = \tilde{I}_{q-1}^q \mathbf{u}_{q-1}^\star$;
the initial solution is the interpolation of the best solution on grid $q-1$.

(1.3)(P) Initialize the Newton iteration number $j := 0$.

Step 2

Compute the initial Jacobi matrix and the defect vector

(2.)(P) $J_q^0 = K_q'[\mathbf{u}_q^0]$; $\quad \mathbf{d}_q^1 = \mathbf{f}_q - K_q\mathbf{u}_q^0$.

Step 3

(3.)(P) Choose a relaxation parameter τ_q^j with $0 < \tau_q^j \le 1$ and a relative accuracy parameter ε_{lin} with $0 < \varepsilon_{\text{lin}} < 1$.

Step 4

(4.)(S) Solve the linear defect system

$$J_q^j \mathbf{w}_q^{j+1} = \mathbf{d}_q^{j+1} \tag{35}$$

approximately (with relative accuracy ε_{lin}) using a PCG solver as described in Subsection 3.3. The result is $\tilde{\mathbf{w}}_q^{j+1}$.

Step 5
Correct the solution:

(5.)(P) $\mathbf{u}_q^{j+1} = \mathbf{u}_q^j + \tau_q^j \tilde{\mathbf{w}}_q^{j+1}.$

Step 6
Control the convergence (parameter c_τ is chosen a priori with $c_\tau < 1$):

(6.1)(P) Compute the new defect vector and the new Jacobi matrix
$\mathbf{d}_q^{j+2} = \mathbf{f}_q - K_q \mathbf{u}_q^{j+1}; \quad J_q^{j+1} = K_q'[\mathbf{u}_q^{j+1}];$

(6.2)(C) Compute defect norms
$d_q^{j+1} = \|\mathbf{d}_q^{j+1}\|; \quad d_q^{j+2} = \|\mathbf{d}_q^{j+2}\|;$

(6.3)(P) $\texttt{IF } d_q^{j+2} \geq d_q^{j+1} \texttt{ THEN } \left(\begin{array}{l} \tau_q^j := \min \left\{ c_\tau \tau_q^j, \dfrac{\tau_q^j d_q^{j+1}}{d_q^{j+1} + d_q^{j+2}} \right\}; \\ \texttt{GOTO } \textit{Step 5} \end{array} \right);$

(6.4)(P) $\texttt{IF } d_q^{j+2} \leq \varepsilon d_q^1 \texttt{ THEN } \left(\begin{array}{l} \mathbf{u}_q^\star := \mathbf{u}_q^{j+1}; \\ \texttt{IF } q < l \texttt{ THEN } (q := q+1; \texttt{GOTO } \textit{Step 1}); \\ \texttt{IF } q = l \texttt{ THEN EXIT}; \end{array} \right);$

(6.5)(P) Perform a further Newton step:
$j := j + 1;$
`GOTO` *Step 3.*

In this description, (P) indicates that the step is performed completely in parallel, i.e., independently on the processors. The solver (S) includes parallel independent parts, communication between processors handling neighbouring subdomains, and global communication, cf. Subsection 3.3. Note, (C) indicates that global communication is necessary. Obviously, the only additional communication (compared with solving a linear problem) is the computation of global defect norms.

4.2 Numerical Results

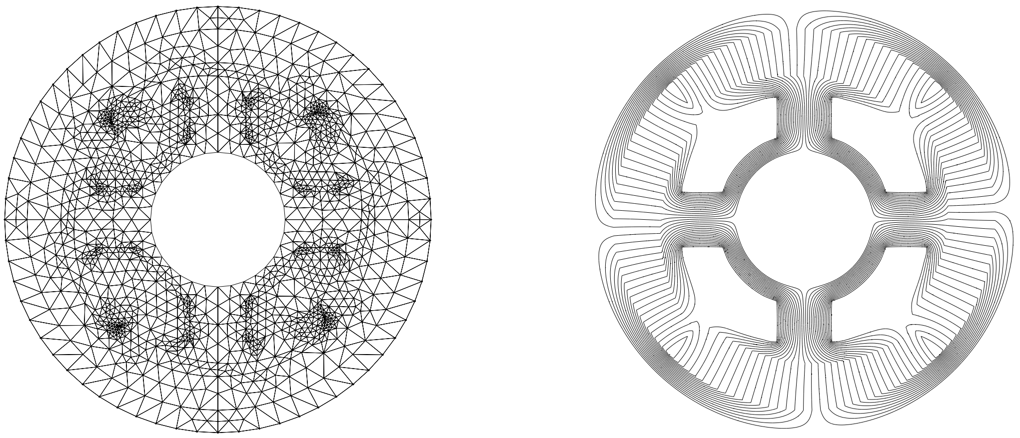

Figure 3: FEM discretization of the electronic motor (coarse grid) and equipotential lines of the solution.

A direct current motor designed for electronic devices (electronic motor) which is excited by permanent magnets serves as a first real-life test example. The interior of the machine is discretized by finite elements (cf. Figure 3). Calculations have been made for both the machine with homogeneous Dirichlet conditions on its boundary and the infinite domain with Sommerfeld's radiation condition where the infinite exterior domain is discretized by BEM. In order to obtain efficiency results, we consider additionally two model problems representing a quarter and a half of the whole machine, i.e. we discretize only sectors of 90 and 180 degrees, respectively, and impose Dirichlet conditions on the boundary.

Example	Dirichlet b. c.	Dirichlet b. c.	Dirichlet b. c.	radiation condition
subdomains	16 FEM	32 FEM	64 FEM	63 FEM 1 BEM
No. of unknowns	374 129	734 199	1 514 008	1 489 416
Newton it. CG iter. 1st grid	4 6,11/16,11	5 6,11/14,11,12	4 6,11/15,11	4 6,11/16,13
CG iter. 2nd grid	9,10	9,11	9,11	9,11
CG iter. 3rd grid	10,12	10,12	10,12	11,13
CG iter. 4th grid	11,13	11,13	11,13	11,14
CG iter. 5th grid	11,15	11,15	11,15	12,15
Newton it. CG iter. 6th grid	4 12,16,9,17	4 12,17,10,16	4 13,16,10,16	4 13,16,10,16
generation linear solver	21.8 63.4	21.6 64.9	23.2 71.7	26.4 85.6
Total time	85.2	86.5	94.9	112.0
Scale-up (norm.) Scaled eff. (rel.)	1.0 $\rightarrow$ 1.0 $\rightarrow$	1.933 $\rightarrow$ 0.966 $\rightarrow$	3.633 0.908	

Time in seconds, scale-up (normalized) and scaled efficiency (relative) on a GC-Power Plus using 16, 32, 64 processors, respectively; 2 Newton iterations on the grids 2–5, relative accuracy $\varepsilon = 10^{-6}$.

Table 2: Performance for a practical problem (electronic motor).

The components of the PNN algorithm are chosen in the standard way [23, 24, 28]. In particular, the parameter ε_{lin} can be adapted to the quadratic convergence speed of the Newton method [24, 30]. Here, a slash (/) marks the change from $\varepsilon_{\text{lin}} = 10^{-2}$ to $\varepsilon_{\text{lin}} = 10^{-4}$ in the accuracy of the CG solver. The components of the PCG solver have been chosen as follows:

$$C_{\Lambda,i} \;:\; \text{Circ } (i \in \mathcal{I}_B) \qquad C_C \;:\; \text{BPS-D}$$
$$C_{I,i} \;:\; \text{V11 } (i \in \mathcal{I}_F) \qquad B_{I,i} \;:\; \text{HExt } (i \in \mathcal{I}_F).$$

The components involved and the numerical effort for nonlinear problems are discussed in [26]. Numerical results are given in Table 2. We present further results, in particular with respect to efficiency, in [30]. Computations with up to 128 processors are documented in [26, 27]. In [28], the application of a parallelized global multigrid method based on DD ideas in Step 4 of the algorithm PNN is discussed. Further, in [29], we demonstrate that the CG with a global BPX or a global multigrid preconditioner yields a robust solver for practical problems.

The second practical example, a technical direct current motor (dc motor, see Subsection 2.3), is to demonstrate the complete algorithm. Starting with a user mesh, we apply an automatic domain decomposition procedure (see Subsection 2.3 and Appendix A) and a parallel mesh generator, the basic ideas of which are presented in [11], to obtain the initial mesh ($q = 1$), which is to be refined four times to get the final mesh for our computations ($q = l = 5$). We present the numerical results in Table 3 and the level lines of the solution in

Example	Dirichlet b. c.	Dirichlet b. c.	radiation condition	radiation condition
Choice for C_C	BPS-D	S-BPX	BPS-D	S-BPX
subdomains	32 FEM	32 FEM	31 FEM 1 BEM	31 FEM 1 BEM
No. of unknowns	417 328	417 328	414 568	414 568
Newton it. 1st grid CG iter. 1st grid	7 16,12,13, 13,13,17,24	6 10,12,12 12,12,11	7 19,14,16, 16,12,19,28	7 10,14,14 14,14,13,17
Newton it. 2nd grid CG iter. 2nd grid	2 16,23	2 13,16	2 18,25	2 15,20
Newton it. 3rd grid CG iter. 3rd grid	2 17,27	2 15,17	2 18,31	2 19,24
Newton it. 4th grid CG iter. 4th grid	2 17,35	2 16,18	2 18,34	2 19,29
Newton it. 5th grid CG iter. 5th grid	4 18,38,22,30	4 16,21,18,20	4 18,42,22,33	4 20,32,27,33
Time (generation) Time (linear solver)	25.2 168.9	25.2 137.8	33.6 178.8	33.6 198.5
Total time	194.1	163.0	212.4	232.1

Time in seconds, GC-Power Plus, 32 processors; relative accuracy $\varepsilon = 10^{-6}$.

Table 3: Performance for the dc motor.

Figure 6. Again, we have done calculations for both, the machine with homogeneous Dirichlet conditions on its boundary and the infinite domain with the radiation condition. Best results with respect to the total computing time have been achieved with $\varepsilon_{\text{lin}} = 0.01$. All other components of the algorithm, except C_C, are chosen as for the electronic motor example.

5 Generalization to Linear Elasticity Problems

5.1 The Mixed Boundary and Domain Integral Variational Formulation

We now want to extend the ideas discussed above to problems of plane linear elasticity in which the displacement $u(x) = (u_1(x), u_2(x))^T$ satisfies formally the system of Lamé equations

$$\begin{array}{l} -\mu(x)\Delta u(x) - (\lambda(x)+\mu(x))\text{grad div}\, u(x) = f(x) \quad \text{in } \Omega \\ u(x) = 0 \text{ on } \Gamma_D, \quad \sum_{l=1}^{2} \sigma_{kl}(u(x))n_l = g_k(x) \text{ on } \Gamma_N, \ (k=1,2) \end{array} \tag{36}$$

where Ω is a bounded Lipschitz domain, $\sigma_{kl}(u)$ are the components of the stress tensor $\sigma(u)$ and $n(x) = (n_1(x), n_2(x))^T$ is the outward normal vector to $\Gamma_D \cup \Gamma_N = \Gamma := \partial\Omega$ ($\Gamma_D \neq \emptyset$) and λ and μ, $\lambda, \mu > 0$, are the Lamé coefficients of the elastic material. In (36), $f = (f_1, f_2)^T$ is the vector of volumic forces, $g = (g_1, g_2)^T$ is the vector of boundary tractions.

As before, Ω is being decomposed into non-overlapping subdomains Ω_i, $i = 1, \ldots, p$, cf. (6). We then have μ_i and λ_i as Lamé coefficients for each of the Ω_i's. Similar to (8) we assume that the volumic forces vanish for $x \in \Omega_i$ with $i \in \mathcal{I}_B$. For simplicity we assume additionally that $\Gamma_N \cap \partial\Omega_i = \emptyset$ for $i \in \mathcal{I}_B$. Then we can write the variational formulation as follows: Find $(\sigma, u) \in \mathbf{V} := \mathbf{\Lambda} \times \mathbf{U}_0$:

$$a(\sigma, u; \tau, v) = \langle F, v\rangle \quad \forall\, (\tau, v) \in \mathbf{V}, \tag{37}$$

where

$$a(\sigma, u; \tau, v) \ := \ a_B(\sigma, u; \tau, v) + a_F(u, v)$$

$$
\begin{aligned}
a_B(\sigma, u; \tau, v) &:= \sum_{i \in \mathcal{I}_B} \nu_i \Big\{ \langle \mathcal{D}_i u_i, v_i \rangle_{\Gamma_i} + \frac{1}{2} \langle \sigma_i, v_i \rangle_{\Gamma_i} + \langle \sigma_i, \mathcal{K}_i v_i \rangle_{\Gamma_i} \\
&\qquad + \langle \tau_i, \mathcal{V}_i \sigma_i \rangle_{\Gamma_i} - \langle \tau_i, \mathcal{K}_i u_i \rangle_{\Gamma_i} - \frac{1}{2} \langle \tau_i, u_i \rangle_{\Gamma_i} \Big\} \\
a_F(u, v) &:= \sum_{i \in \mathcal{I}_F} \int_{\Omega_i} \left(\lambda_i \, div\, u(x)\, div\, v(x) + 2\, \mu_i \sum_{k,l=1}^{2} \epsilon_{kl}(u)\, \epsilon_{kl}(v) \right) dx \\
\langle F, v \rangle &:= \sum_{i \in \mathcal{I}_F} \int_{\Omega_i} f(x)\, v(x)\, dx + \int_{\Gamma_N} g(x)\, v(x)\, ds,
\end{aligned}
$$

with the duality pairing $\langle ., . \rangle$, the traces $u_i = u|_{\partial \Omega_i}$ and the boundary tractions $\sigma = [\sigma_i]_{i \in \mathcal{I}_B}$ belonging to $\partial \Omega_i$ and the strain $\epsilon_{kl}(u) := (\partial u_k / \partial x_l + \partial u_l / \partial x_k)/2$. Note, for simplicity we have assumed that Ω is bounded such that, in contrary to Section 3.1, special terms for the exterior domain do not occur in the definition of $a_B(.;.)$. Nevertheless, the ideas presented above concerning exterior problems can be applied analogously.

Let the spaces $\mathbf{U}_0$ and $\mathbf{\Lambda}$ be defined as follows:

$$
\begin{aligned}
\mathbf{U}_0 &:= \{ u \in [\mathrm{H}^1(\Omega)]^2 : \; u|_{\Gamma_{BE}} \in \mathrm{H}^{1/2}(\Gamma_{BE}), u|_{\Gamma_D} = 0 \} \\
\mathbf{\Lambda} &:= \textstyle\prod_{i \in \mathcal{I}_B} \left[\mathrm{H}^{-1/2}(\partial \Omega_i) \right]^2,
\end{aligned}
$$

where $\Gamma_{BE} := \bigcup_{i \in \mathcal{I}_B} \partial \Omega_i \setminus \Gamma_D, \Gamma_{FE} := \bigcup_{i \in \mathcal{I}_F} \partial \Omega_i \setminus \Gamma_D, \Gamma_C := \Gamma_{BE} \cup \Gamma_{FE}$. Let $\Omega_F := \cup_{i \in \mathcal{I}_F} \Omega_i$, then we consider the following norm in $\mathbf{V}$:

$$
\|(\sigma, u)\|_{\mathbf{V}} := (\|\sigma\|^2_{\mathbf{\Lambda}} + \|u\|^2_{\mathrm{H}^{1/2}(\Gamma_{BE})} + \|u\|^2_{\mathrm{H}^1(\Omega_F)})^{1/2}. \tag{38}
$$

The boundary integral operators $\mathcal{V}_i$, $\mathcal{K}_i$ and $\mathcal{D}_i$ are defined as in (11), where $\partial(.)$ has to be replaced by the operator $\mathcal{T}(.)$ which is defined in its strong form as $\mathcal{T}(.) := 2\mu\partial(.) + \lambda n\, div\, u + \mu\, n \times curl(.)$ and in its weak form by the first Green formula. For $\mathcal{E}(x, y)$ we now have to insert the well known Kelvin fundamental solution (see, e.g., [6]). The variational formulation (37) has a unique solution provided that the single layer potential operators $\mathcal{V}_i$ are positive definite ($\mathrm{H}^{-1/2}(\partial \Omega_i)$–elliptic) [6, 32, 40].

Similar to Section 3.2 we discretize (37) to obtain a system of equations which is manipulated in the same way as discussed in Section 3.3.1. This leads again to a symmetric, positive definite system matrix. Thus, the parallel solution can be performed in a similar fashion provided suitable preconditioners C_Λ, C_C, C_I and B_I are known (see Section 5.2 and [40]).

5.2 Numerical Results

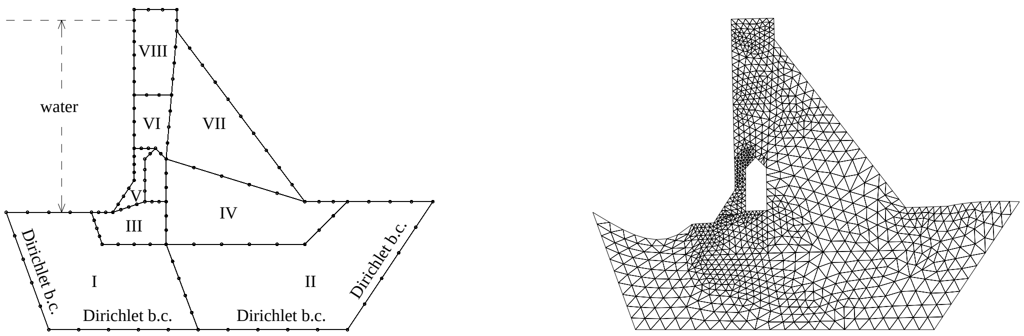

Figure 4: The subdomains and the BE discretization of the 1^{st} level (left) and the deformed (magnification factor 100) FE grid of the 2^{nd} level (right).

As a test problem we consider a model of dam filled with water as sketched in Figure 4. As indicated there, boundary conditions are given on Γ_D (zero displacement) and on Γ_N (zero or

	BEM: I-VIII		FEM: III-VIII BEM: I-II		FEM: I-VIII	
l	I(ε)	CPU	I(ε)	CPU	I(ε)	CPU
2	27	4.9	25	4.6	25	3.9
3	27	8.8	28	8.8	28	7.9
4	28	24.1	30	27.9	31	22.8
5	29	85.9	31	104.9	32	81.3
s(5)	49.5		68.5		63.8	
N(5)	6470		78130		119318	

Table 4: Levels (l), pure solution time (s), number of unknowns (N), iteration count ($I(\varepsilon)$, $\varepsilon = 10^{-6}$), CPU time in seconds for the dam-problem. The experiments were carried out on a Power-XPlorer using 8 processors.

according to the water pressure, respectively). The Lamé constants are given for rock (I-II) by $\mu_r = 7.265\mathrm{e}4\mathrm{MPa}$, $\lambda_r = 3.743\mathrm{e}4\mathrm{MPa}$ and for concrete (III-VIII) by $\mu_c = 9.2\mathrm{e}6\mathrm{MPa}$, $\lambda_c = 9.2\mathrm{e}6\mathrm{MPa}$. For the results presented in Table 4, the operators C_Λ, C_C, C_I, B_I have been chosen as follows:

$$C_{\Lambda,i} \ : \ \text{mgV} \ (i \in \mathcal{I}_B) \qquad\qquad C_C \ : \ \text{mgD}$$
$$C_{I,i} \ : \ \text{V11(HExt+S)} \ (i \in \mathcal{I}_F) \qquad\qquad B_{I,i} \ : \ \text{HExt+S} \ (i \in \mathcal{I}_F).$$

Here, (mgV) stands for a multigrid-based preconditioner for the single layer potential (see [1]). Furthermore, new algorithms for $C_{I,i}$ and $B_{I,i}$ have been used. In particular, the hierarchical extension (HExt+S) has been improved by a coarse grid solver and smoothing on the other levels [13].

The BE discretization of the 1^{st} level and the FE discretization of the 2^{nd} level (deformed mesh) are shown in Figure 4. We use piecewise linear trial functions for the displacements and piecewise constants for the boundary tractions. The entries of the BE matrices are computed fully analytically.

In Table 4 we present several combinations of FE/BE discretizations. Looking at the CPU-time we observe that the FE discretization (column 3) leads to the best results. However, if we are interested in the pure solving time $s(.)$ ($s(5)$ for the 5^{th} level is given in Table 4) the BE discretization (first column) is of advantage.

6 Concluding Remarks and Generalizations

The DD-method has turned out to be a powerful tool for establishing the coupled FE/BE variational formulation and for solving the discrete systems efficiently on massively parallel computers. The results presented here have been obtained using the code FEM⊗BEM [14] which can solve linear and non-linear magnetic field problems as well as problems arising in linear elasticity. The high efficiency and the scalability of the algorithm has been demonstrated [5, 16, 30, 26, 28, 36].

Comparison of "local" DD methods described in this paper with "global" multigrid methods implemented on massively parallel machines as well as workstation clusters is given in [28]. The use and the parallelization of "global" methods is also discussed in [4]. Other coupling and solution techniques are studied in [10, 41, 43].

The techniques presented here can be generalized to the 3D case provided that fast matrix-by-vector multiplication routines for the BE matrices, e.g. based on Panel-Clustering-Techniques developed in [22], and asymptotically optimal, or almost optimal components C_I (e.g. multigrid preconditioners), B_I [38], C_C (e.g. BPX) and C_Λ [35, 39, 40] of the preconditioners are available.

References

[1] J. H. Bramble, Z. Leyk, and J. E. Pasciak. The analysis of multigrid algorithms for pseudodifferential operators of order minus one. *Math. Comp.*, 63:461–478, 1994.

[2] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation*, 50(181):1–17, 1988.

[3] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. The construction of preconditioners for elliptic problems by substructuring I – IV. *Mathematics of Computation*, 1986, 1987, 1988, 1989. 47, 103–134, 49, 1–16, 51, 415–430, 53, 1–24.

[4] U. Brink, M. Kreienmeyer, and E. Stein. Different methodologies for coupled BEM and FEM with implementation on parallel computers. In Reports from the Final Conference of the Priority Research Programme Boundary Element Methods 1989–1995, (W. Wendland ed.), Stuttgart, October 1995, Berlin, 1996, Springer Verlag.

[5] C. Carstensen, M. Kuhn, and U. Langer. Fast parallel solvers for symmetric boundary element domain decomposition equations. Report 499, Johannes Kepler University Linz, Institute of Mathematics, 1995.

[6] C. Carstensen and P. Wriggers. On the symmetric boundary element method and the symmetric coupling of boundary elements and finite elements. *IMA Journal of Numerical Analysis*, 1996. Submitted.

[7] M. Costabel. Symmetric methods for the coupling of finite elements and boundary elements. In C. A. Brebbia, W. L. Wendland, and G. Kuhn, editors, *Boundary Elements IX*, pages 411–420. Springer–Verlag, 1987.

[8] M. Costabel. Boundary integral operators on Lipschitz domains: Elementary results. *SIAM J. Math. Anal.*, 19:613–626, 1988.

[9] M. Dryja. A capacitance matrix method for Dirichlet problems on polygonal regions. *Numerische Mathematik*, 39(1):51–64, 1982.

[10] S. A. Funken and E. P. Stephan. Fast solvers with multigrid preconditioners for linear FEM–BEM coupling. Preprint, Universität Hannover, Institut für Angewandte Mathematik, Hannover, 1995.

[11] G. Globisch. PARMESH - a parallel mesh generator. *Parallel Computing*, 21(3):509–524, 1995.

[12] M. Goppold, G. Haase, B. Heise, and M. Kuhn. Preprocessing in BE/FE domain decomposition methods. Technical Report, Johannes Kepler University Linz, 1996. In Preparation.

[13] G. Haase. Hierarchical extension operators plus smoothing in domain decomposition preconditioners. Report 497, Johannes Kepler University Linz, Institute of Mathematics, 1995.

[14] G. Haase, B. Heise, M. Jung, and M. Kuhn. FEM⊗BEM - a parallel solver for linear and nonlinear coupled FE/BE-equations. DFG-Schwerpunkt "Randelementmethoden", Report 94-16, University Stuttgart, 1994.

[15] G. Haase and U. Langer. The non-overlapping domain decomposition multiplicative Schwarz method. *International Journal of Computer Mathematics*, 44:223–242, 1992.

[16] G. Haase and U. Langer. The efficient parallel solution of PDEs. *Computers Math. Applic.*, 1996. To appear.

[17] G. Haase, U. Langer, and A. Meyer. A new approach to the Dirichlet domain decomposition method. In S. Hengst, editor, *Fifth Multigrid Seminar, Eberswalde 1990*, pages 1–59, Berlin, 1990. Karl–Weierstrass–Institut. Report R–MATH–09/90.

[18] G. Haase, U. Langer, and A. Meyer. The approximate dirichlet domain decomposition method. Part I: An algebraic approach. Part II: Applications to 2nd-order elliptic boundary value problems. *Computing*, 47:137–151 (Part I), 153–167 (Part II), 1991.

[19] G. Haase, U. Langer, and A. Meyer. Domain decomposition preconditioners with inexact subdomain solvers. *J. of Num. Lin. Alg. with Appl.*, 1:27–42, 1992.

[20] G. Haase, U. Langer, A. Meyer, and S. V. Nepomnyaschikh. Hierarchical extension operators and local multigrid methods in domain decomposition preconditioners. *East-West J. Numer. Math.*, 2(3):173–193, 1994.

[21] W. Hackbusch. *Multi–Grid methods and applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer–Verlag, Berlin, 1985.

[22] W. Hackbusch, L. Lage, and S. A. Sauter. On the efficient realization of sparse matrix techniques for integral equations with focus on panel clustering, cubature and software design aspects. In Reports from the Final Conference of the Priority Research Programme Boundary Element Methods 1989–1995, (W. Wendland ed.), Stuttgart, October 1995, Berlin, 1996, Springer Verlag.

[23] B. Heise. Mehrgitter–Newton–Verfahren zur Berechnung nichtlinearer magnetischer Felder. Wissenschaftliche Schriftenreihe 4/1991, Technische Universität Chemnitz, 1991.

[24] B. Heise. Nonlinear field calculations with multigrid-Newton methods. *IMPACT of Computing in Science and Engineering*, 5:75–110, 1993.

[25] B. Heise. Analysis of a fully discrete finite element method for a nonlinear magnetic field problem. *SIAM J. Numer. Anal.*, 31(3):745–759, 1994.

[26] B. Heise. Nonlinear field simulation with FE domain decomposition methods on massively parallel computers. 1995. Submitted to *Surveys on Mathematics for Industry.*

[27] B. Heise. Nonlinear simulation of electromagnetic fields with domain decomposition methods on MIMD parallel computers. *Journal of Computational and Applied Mathematics*, 62, 1995. To appear.

[28] B. Heise and M. Jung. Comparison of parallel solvers for nonlinear elliptic problems based on domain decomposition ideas. Report 494, Johannes Kepler University Linz, Institute of Mathematics, 1995.

[29] B. Heise and M. Jung. Robust parallel multilevel methods. 1995. Submitted for Publication.

[30] B. Heise and M. Kuhn. Parallel solvers for linear and nonlinear exterior magnetic field problems based upon coupled FE/BE formulations. *Computing*, 1996. To appear. Also Report No. 486, Inst. of Mathematics, Univ. Linz, 1995.

[31] S. Holzer. On the engineering analysis of 2D problems by the symmetric Galerkin boundary element method and coupled BEM/FEM. In J. H. Kane, G. Maier, N. Tosaka, and S. N. Atluri, editors, *Advances in boundary element techniques*, Berlin / New York / Heidelberg, 1992. Springer.

[32] G. C. Hsiao, B. N. Khoromskij, and W. L. Wendland. Boundary integral operators and domain decomposition. Preprint 94–11, Universität Stuttgart, Mathematisches Institut A, Stuttgart, 1994.

[33] G. C. Hsiao and W. L. Wendland. Domain decomposition in boundary element methods. In *Proc. of IV Int. Symposium on Domain Decomposition Methods, (R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, O. B. Widlund eds.), Moscow, May 1990*, pages 41–49, Philadelphia, 1991. SIAM Publ.

[34] B. N. Khoromskij, G. E. Mazurkevich, and E. P. Zhidkov. Domain decomposition method for magnetostatics nonlinear problems in combined formulation. *Sov. J. Numer. Anal. Math. Modelling*, 5(2):111–136, 1990.

[35] B. N. Khoromskij and W. L. Wendland. Spectrally equivalent preconditioners for boundary equations in substructuring techniques. *East–West Journal of Numerical Mathematics*, 1(1):1–26, 1992.

[36] M. Kuhn and U. Langer. Parallel algorithms for symmetric boundary element equations. *ZAMM*, 1995. Submitted for publication.

[37] U. Langer. Parallel iterative solution of symmetric coupled FE/BE- equations via domain decomposition. *Contemporary Mathematics*, 157:335–344, 1994.

[38] S. V. Nepomnyaschikh. Optimal multilevel extension operators. Preprint SPC 95–3, Technische Universität Chemnitz–Zwickau, Fakultät für Mathematik, 1995.

[39] S. Rjasanow. Vorkonditionierte iterative Auflösung von Randelementgleichungen für die Dirichlet–Aufgabe. Wissenschaftliche Schriftenreihe 7/1990, Technische Universität Chemnitz, 1990.

[40] S. Rjasanow. Optimal preconditioner for boundary element formulation of the dirichlet problem in elasticity. *Math. Methods in the Applied Sciences*, 18:603–613, 1995.

[41] E. Schnack and K. Türke. Coupling of FEM and BEM for elastic structures. In Reports from the Final Conference of the Priority Research Programme Boundary Element Methods 1989–1995, (W. Wendland ed.), Stuttgart, October 1995, Berlin, 1996, Springer Verlag.

[42] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Comput. System in Eng.*, 2:135–148, 1991.

[43] O. Steinbach and W. L. Wendland. Efficient preconditioners for boundary element methods and their use in domain decomposition. DFG-Schwerpunkt "Randelementmethoden", Report 95-19, University Stuttgart, 1995.

[44] C. H. Tong, T. F. Chan, and C. J. Kuo. A domain decomposition preconditioner based on a change to a multilevel nodal basis. *SIAM J. Sci. Stat. Comput.*, 12(6):1486–1495, 1991.

A The Direct Current Motor

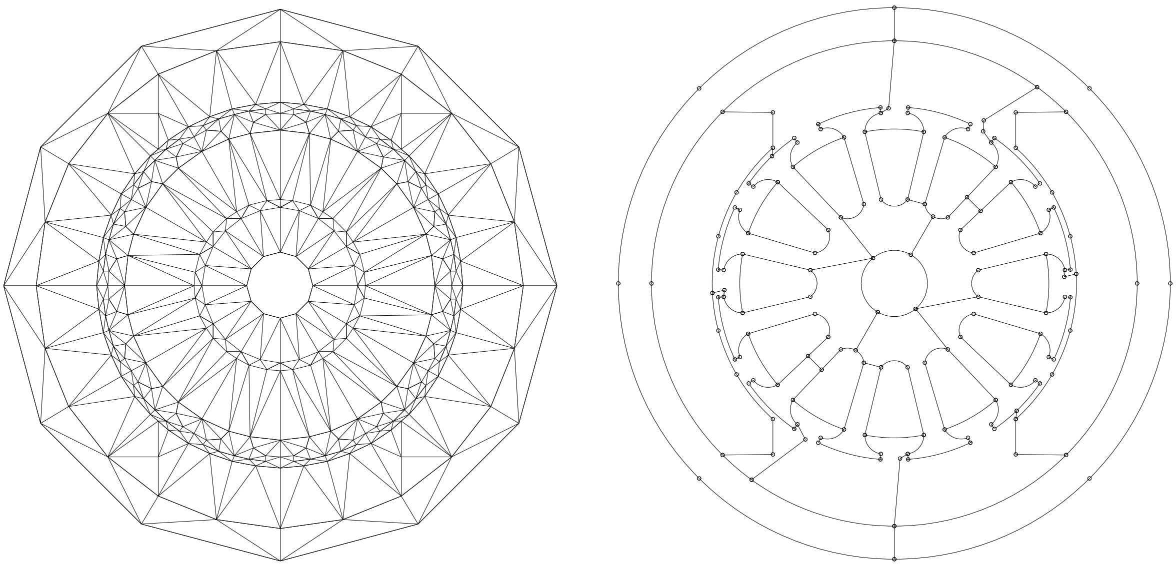

Figure 5: Initial triangular-based description and final decomposition of the dc-motor into 32 subdomains.

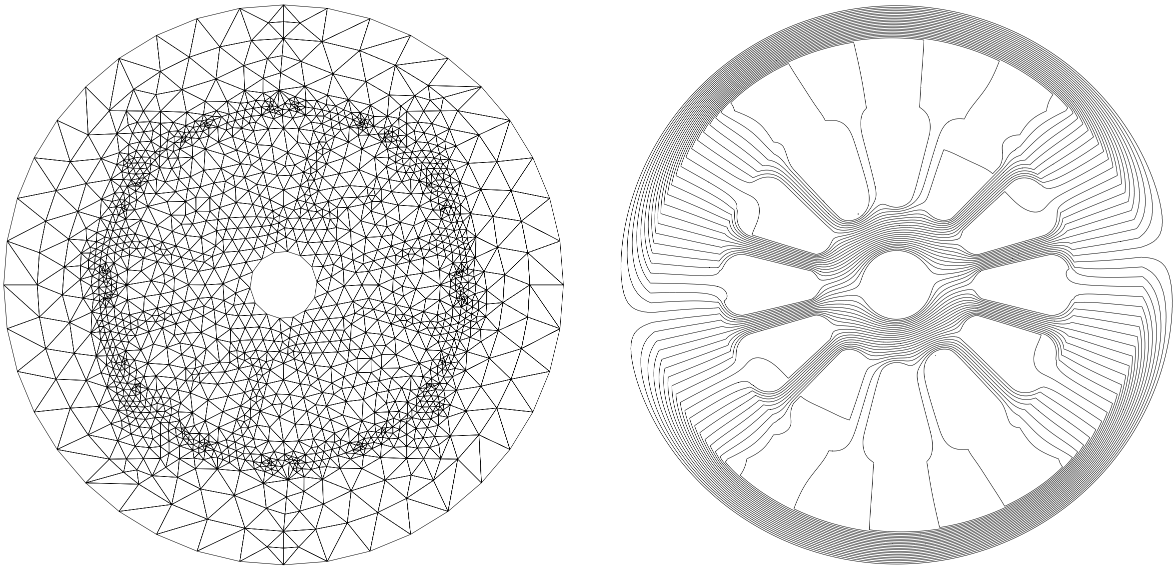

Figure 6: The mesh of the 1^{st} level and the equipotential lines of the solution (dc-motor).

B The Parallel Algorithm

	FEM ($i \in \mathcal{I}_F$)	BEM ($i \in \mathcal{I}_B$)
0.	Starting step	
	Choose an initial guess $\mathbf{u} = \mathbf{u}_0$	
	$\mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{C,i} \\ \mathbf{u}_{I,i} \end{bmatrix}$	$\mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{\Lambda,i} \\ \mathbf{u}_{C,i} \end{bmatrix}$
	$\mathbf{r}_{I,i} = \mathbf{f}_{I,i} - K_{IC,i}\mathbf{u}_{C,i} - K_{I,i}\mathbf{u}_{I,i}$ $\mathbf{r}_{C,i} = \mathbf{f}_{C,i} - K_{C,i}\mathbf{u}_{C,i} - K_{CI,i}\mathbf{u}_{I,i}$	$\mathbf{v}_{\Lambda,i} = \mathbf{f}_{\Lambda,i} - K_{\Lambda,i}\mathbf{u}_{\Lambda,i} + K_{\Lambda C,i}\mathbf{u}_{C,i}$ $\mathbf{r}_{\Lambda,i} = C_{\Lambda,i}^{-1}\mathbf{v}_{\Lambda,i}$ $\mathbf{r}_{C,i} = \mathbf{f}_{C,i} - K_{C,i}\mathbf{u}_{C,i} - K_{C\Lambda,i}\mathbf{u}_{\Lambda,i}$ $\mathbf{r}_{C,i} = \mathbf{r}_{C,i} - K_{C\Lambda,i}\mathbf{r}_{\Lambda,i}$
	$\mathbf{v}_{C,i} = \mathbf{r}_{C,i} - K_{CI,i}B_{I,i}^{-T}\mathbf{r}_{I,i}$	$\mathbf{w}_{\Lambda,i} = \mathbf{r}_{\Lambda,i}, \quad \mathbf{p}_i = \mathbf{v}_{\Lambda,i}$ $\mathbf{v}_{C,i} = \mathbf{r}_{C,i}, \quad \mathbf{z}_{\Lambda,i} = K_{\Lambda,i}\mathbf{w}_{\Lambda,i}$
	$\mathbf{w}_{C,i} = A_{C,i}\mathbf{w}_C \leftarrow \mathbf{w}_C = C_C^{-1}\sum_{i=1}^{p} A_{C,i}^T\mathbf{v}_{C,i} \rightarrow \mathbf{w}_{C,i} = A_{C,i}\mathbf{w}_C$	
	$\mathbf{w}_{I,i} = C_{I,i}^{-1}\mathbf{r}_{I,i} - B_{I,i}^{-1}K_{IC,i}\mathbf{w}_{C,i}$	$\mathbf{z}_{\Lambda C,i} = K_{\Lambda C,i}\mathbf{w}_{C,i}, \quad \mathbf{z}_{C,i} = K_{C,i}\mathbf{w}_{C,i}$
	$\mathbf{s} = \mathbf{w}$	$\mathbf{s} = \mathbf{w}$
	$\sigma_i = \mathbf{r}_{C,i}^T\mathbf{w}_{C,i} + \mathbf{r}_{I,i}^T\mathbf{w}_{I,i}$	$\sigma_i = \mathbf{r}_{C,i}^T\mathbf{w}_{C,i} + \mathbf{w}_{\Lambda,i}^T(\mathbf{z}_{\Lambda,i} - \mathbf{p}_i)$
	$\sigma = \sigma^0 = \sum_{i=1}^{p}\sigma_i$	
	Iteration	
1.	$\mathbf{v}_{I,i} = K_{IC,i}\mathbf{s}_{C,i} + K_{I,i}\mathbf{s}_{I,i}$ $\mathbf{v}_{C,i} = K_{C,i}\mathbf{s}_{C,i} + K_{CI,i}\mathbf{s}_{I,i}$ $\delta_i = \mathbf{v}_C^T\mathbf{s}_C + \mathbf{v}_{I,i}^T\mathbf{s}_{I,i}$	$\mathbf{w}_{\Lambda,i} = \mathbf{z}_{\Lambda,i} - \mathbf{z}_{\Lambda C,i}$ $\mathbf{v}_{\Lambda,i} = C_{\Lambda,i}^{-1}\mathbf{w}_{\Lambda,i}$ $\mathbf{v}_{C,i} = \mathbf{z}_{C,i} + K_{C\Lambda,i}(\mathbf{s}_{\Lambda,i} - \mathbf{v}_{\Lambda,i})$ $\delta_i = \mathbf{v}_{C,i}^T\mathbf{s}_{C,i} + \mathbf{v}_{\Lambda,i}^T\mathbf{z}_{\Lambda,i} - \mathbf{w}_{\Lambda,i}^T\mathbf{s}_{\Lambda,i}$
	$\delta = \sum_{i=1}^{p}\delta_i$	
	$\alpha = \sigma/\delta$	$\alpha = \sigma/\delta$
2.	$\hat{\mathbf{u}}_i = \mathbf{u}_i + \alpha\mathbf{s}_i$ $\hat{\mathbf{r}}_i = \mathbf{r}_i - \alpha\mathbf{v}_i$	$\hat{\mathbf{u}}_i = \mathbf{u}_i + \alpha\mathbf{s}_i$ $\hat{\mathbf{r}}_i = \mathbf{r}_i - \alpha\mathbf{v}_i$
3.	$\hat{\mathbf{v}}_{C,i} = \hat{\mathbf{r}}_{C,i} - K_{CI,i}B_{I,i}^{-T}\hat{\mathbf{r}}_{I,i}$	$\hat{\mathbf{v}}_{C,i} = \hat{\mathbf{r}}_{C,i}, \quad \mathbf{h}_{\Lambda,i} = K_{\Lambda,i}\hat{\mathbf{r}}_{\Lambda,i}$
	$\mathbf{w}_{C,i} = A_{C,i}\mathbf{w}_C \leftarrow \mathbf{w}_C = C_C^{-1}\sum_{i=1}^{p} A_{C,i}^T\mathbf{v}_{C,i} \rightarrow \mathbf{w}_{C,i} = A_{C,i}\mathbf{w}_C$	
	$\hat{\mathbf{w}}_{I,i} = C_{I,i}^{-1}\hat{\mathbf{r}}_{I,i} - B_{I,i}^{-1}K_{IC,i}\hat{\mathbf{w}}_{C,i}$	$\hat{\mathbf{w}}_{\Lambda,i} = \hat{\mathbf{r}}_{\Lambda,i}, \quad \hat{\mathbf{p}}_i = \mathbf{p}_i - \alpha\mathbf{w}_{\Lambda,i}$ $\mathbf{h}_{\Lambda C,i} = K_{\Lambda C,i}\hat{\mathbf{w}}_{C,i}$ $\mathbf{h}_{C,i} = K_{C,i}\hat{\mathbf{w}}_{C,i}$
4.	$\sigma_i = \hat{\mathbf{r}}_{C,i}^T\hat{\mathbf{w}}_{C,i} + \hat{\mathbf{r}}_{I,i}^T\hat{\mathbf{w}}_{I,i}$	$\sigma_i = \hat{\mathbf{r}}_{C,i}^T\hat{\mathbf{w}}_{C,i} + \mathbf{h}_{\Lambda,i}^T\hat{\mathbf{r}}_{\Lambda,i} - \hat{\mathbf{p}}_i^T\hat{\mathbf{r}}_{\Lambda,i}$
	$\hat{\sigma} = \sum_{i=1}^{p}\sigma_i$	
	$\beta = \hat{\sigma}/\sigma$	$\beta = \hat{\sigma}/\sigma$
5.	$\hat{\mathbf{s}}_i = \hat{\mathbf{w}}_i + \beta\mathbf{s}_i$	$\hat{\mathbf{s}}_i = \hat{\mathbf{w}}_i + \beta\mathbf{s}_i, \quad \hat{\mathbf{z}}_i = \mathbf{h}_i + \beta\mathbf{z}_i$
6.	If $\hat{\sigma} \leq \varepsilon^2 * \sigma^0$, then STOP else goto step 1.	