Technische Universität Graz

TUG

Inexact Data–Sparse Boundary Element Tearing and Interconnecting Methods

U. Langer, G. Of, O. Steinbach, W. Zulehner

Berichte aus dem Institut für Mathematik D Numerik und Partielle Differentialgleichungen

Bericht 2005/4

Technische Universität Graz

Inexact Data–Sparse Boundary Element Tearing and Interconnecting Methods

U. Langer, G. Of, O. Steinbach, W. Zulehner

Berichte aus dem
Institut für Mathematik
Numerik und Partielle Differentialgleichungen

Bericht 2005/4

Technische Universität Graz
Institut für Mathematik D (Numerik und Partielle Differentialgleichungen)
Steyrergasse 30
A 8010 Graz

WWW: `http://www.numerik.math.tu-graz.at`

Inexact Data–Sparse Boundary Element Tearing and Interconnecting Methods*

U. Langer[1,2], G. Of[3], O. Steinbach[4], W. Zulehner[2]

[1]Radon Institute for Computational and Applied Mathematics,
Austrian Academy of Sciences,
Altenberger Strasse 69, A–4040 Linz, Austria

[2]Institute for Computational Mathematics,
Johannes Kepler University of Linz,
Altenberger Strasse 69, A–4040 Linz, Austria

[3]Institute for Applied Analysis and Numerical Simulation,
University of Stuttgart,
Pfaffenwaldring 57, D–70569 Stuttgart, Germany

[4]Institute for Mathematics D,
Technical University of Graz,
Steyrergasse 30, A–8010 Graz, Austria

Abstract

The Boundary Element Tearing and Interconnecting (BETI) methods have recently been introduced as boundary element counterparts of the well–established Finite Element Tearing and Interconnecting (FETI) methods. In this paper we present inexact data–sparse versions of the BETI methods which avoid the elimination of the primal unknowns and dense matrices. However, instead of symmetric and positive definite systems, we finally have to solve two–fold saddle point problems. The proposed iterative solvers and preconditioners result in almost optimal solvers the complexity of which is proportional to the number of unknowns on the skeleton up to some polylogarithmical factor. Moreover, the solvers are robust with respect to large coefficient jumps.

1 Introduction

The Finite Element Tearing and Interconnecting (FETI) methods were introduced by Farhat and Roux [16] in 1991 as some kind of a dual version of the classical iterative substructuring methods (see also the survey paper [17] by the same authors). Meanwhile the classical FETI methods and, in particular, the more recently developed dual–primal FETI (FETI–DP) methods are well established as powerful and robust parallel solvers for large–scale finite element equations in different fields of applications, see, e.g., [13, 14, 15, 18, 50, 52]. The practical success of the FETI and FETI–DP methods in real–life applications is supported by a rigorous theoretical analysis [7, 34, 35, 40, 41]. In particular, for second–order self–adjoint elliptic problems such as the

*This work has been supported by the Radon Institute for Computational and Applied Mathematics (RICAM) of the Austrian Academy of Sciences, the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' under grant P14953, and the German Research Foundation 'Deutsche Forschungsgemeinschaft (DFG)' under the grant SFB 404 'Multifield Problems in Continuum Mechanics'.

potential equation or the linear elasticity problem, the iteration numbers of the conjugate gradient solver preconditioned by a properly scaled FETI preconditioner grow only like $\mathcal{O}(1 + \log(H/h))$, where H and h denote the usual scaling of the subdomains and the finite elements, respectively. Furthermore, the iteration process is insensitive to large jumps in the coefficients of the partial differential equations. The latter property is very important for many practical applications, e.g. in electromagnetics, where we frequently have to solve interface problems with large jumps in the material coefficients. We refer the reader to the recently published monograph [58] by Toselli and Widlund for more informations about the analysis of FETI methods.

Recently, the Boundary Element Tearing and Interconnecting (BETI) methods [38] and the coupled BETI/FETI methods [39] were introduced. It is widely recognized that Boundary Element Methods (BEM) have some advantages over the Finite Element Methods (FEM) in the treatment of unbounded regions. But we can also benefit from the BEM in some other situations like in the case of large air subdomains without sources in electromagnetics or in the case of moving parts or interfaces in magnetomechanics [32]. The coupling of BEM and FEM within a domain decomposition framework seems to be very attractive (see e.g. [23] and [54]). The symmetric FEM–BEM coupling technique was originally proposed by Costabel [10]. The symmetric coupling technique was then used by Hsiao and Wendland to construct a (primal) boundary element (BE) substructuring method [31], see also [30]. Similar to the finite element substructuring method they eliminate the interior subdomain unknowns, which are the fluxes (tractions) in the BEM, and arrive at the symmetric and positive definite boundary element Schur complement system that can be solved by a direct or an iterative method. For large–scale problems, a preconditioned conjugate gradient method should be used for efficiency reasons. However, in every iteration step, the interior boundary element subdomain unknowns must be eliminated, i.e. a discrete single layer potential must be inverted to compute the solution of a Dirichlet boundary value problem in every subdomain. To avoid the expensive elimination of these interior boundary element unknowns, an inexact boundary element substructuring method was introduced in [36] which requires the solution of a saddle point problem. The saddle point problem was solved by the preconditioned conjugate gradient method proposed by Bramble and Pasciak for solving system of algebraic equations with symmetric, but indefinite system matrices [5]. The availability of good preconditioners is very essential for the efficiency of the solver. In the inexact BE substructuring case preconditioners for the local (subdomain) discrete single layer potential operators and for the global (skeleton) boundary element Schur complement are needed. The latter usually provides the global information exchange. In [9] such preconditioners are proposed and analyzed. Inexact finite element substructuring solvers were proposed and investigated in [4, 24, 25, 26]. The first inexact FETI solver was introduced and analyzed by Klawonn and Widlund in [33]. Let us mention that inexact versions are usually more efficient than Schur complement techniques, especially in the case of sufficiently large local problems. This is typical for really large scale problems. On PC clusters, also mid–size problems can benefit from inexact solution techniques.

In this paper we introduce inexact BETI methods which lead to three–fold saddle point problems in the first instance. However, applying an appropriate projection, we can reduce the three–fold saddle point problem to a two–fold saddle point problem. Following the approach given in [61] we present preconditioned Krylov subspace solution methods for two–fold saddle point problems and give sharp convergence rate estimates. Similar questions arise when considering the coupling of symmetric boundary element with mixed finite element methods [20]. The standard boundary element discretization of boundary integral operators with nonlocal kernel functions would lead to fully populated matrices. This is totally unacceptable for 3D boundary value problems. Indeed, already the matrix–by–vector multiplication costs $\mathcal{O}(N_h^2)$ arithmetical operations in the case of dense matrices. Here N_h denotes the number of boundary unknowns which is of the order $\mathcal{O}(h^{-(d-1)})$, where h denotes the usual mesh size parameter and d is the spatial dimension ($d = 2$, or $d = 3$). The same complexity is required for the storage demand. Data–sparse approximations of the system matrix, such as multipole techniques [8, 51], panel clustering methods [28], $\mathcal{H}$–matrix approaches [27] and Adaptive Cross Approximation (ACA) methods [1, 2, 3], can reduce the complexity to almost $\mathcal{O}(N_h)$, up to polylogarithmic perturbations, for both the arithmetical expenses and the memory demand. Here we use the Fast Multipole Method, developed in [46], in order to approximate the single layer potential, the double layer potential and the hypersingular boundary integral operators which appear in the symmetric boundary integral domain decomposition formulation. Our preconditioned Krylov subspace solvers for two–fold saddle point problems require appropriately scaled preconditioners for the local discrete single layer potential operators, for the local boundary element Schur complements and for the BETI matrix in their data–sparse (fast multipole) representations. We propose data–sparse preconditioners which result in an almost optimal solver requiring $\mathcal{O}((H/h)^{(d-1)}(1+\log(H/h))^s \log\varepsilon^{-1})$ arithmetical operations in a parallel regime and $\mathcal{O}((H/h)^{(d-1)}(1+\log(H/h))^2)$ storage units per processor, where $s \in \{3,4,5\}$ depending on the quality of the preconditioners for the local problems, and $\varepsilon \in (0,1)$ is the relative accuracy of the iteration error in a suitable norm. H and h denote the usual scalings of the subdomains and the boundary elements, respectively. Moreover, our solvers are insensitive to large jumps in the coefficients of the potential equation that is considered as model problem throughout the paper.

The rest of the paper is organized as follows: In Section 2, we introduce the data–sparse symmetric boundary element domain decomposition method. Section 3 is devoted to the inexact BETI approach that results in a two–fold saddle–point problem after a suitable subspace projection. In Section 4, we present and analyze iterative methods for solving two–fold saddle–point problems. Optimal data–sparse boundary element preconditioners required by the solvers presented in Section 4 are given in Section 5. This completes the solution procedure. Finally, we discuss some numerical results in Section 6 and draw some conclusions in Section 7.

2 Data–Sparse Boundary Element Domain Decomposition Methods

As a model problem we consider the Dirichlet boundary value problem for the potential equation

$$-\mathrm{div}[\alpha(x)\nabla\hat{u}(x)] = 0 \text{ for } x \in \Omega \subset I\!R^3, \quad \hat{u}(x) = g(x) \text{ for } x \in \Gamma = \partial\Omega, \tag{2.1}$$

where Ω is a bounded Lipschitz domain that is assumed to be decomposed into p non–overlapping subdomains Ω_i with Lipschitz boundaries $\Gamma_i = \partial\Omega_i$. We further assume that the coefficient function $\alpha(\cdot)$ in the potential equation (2.1) is piecewise constant such that $\alpha(x) = \alpha_i > 0$ for $x \in \Omega_i,\ i = 1,\dots,p$. The variational formulation of the boundary value problem (2.1) is to find $\hat{u} \in H^1(\Omega)$ with $\hat{u}_{|\Gamma} = g$ such that the variational equation

$$\sum_{i=1}^{p} \alpha_i \int\limits_{\Omega_i} \nabla\hat{u}(x)\,\nabla v(x)dx \;=\; 0 \tag{2.2}$$

is satisfied for all test functions $v \in H_0^1(\Omega)$, where g is a given function from $H^{1/2}(\Gamma)$. Applying Green's first formula with respect to the local subdomains Ω_i gives

$$\sum_{i=1}^{p} \alpha_i \int\limits_{\Gamma_i} \frac{\partial}{\partial n_i}\hat{u}(x)v(x)ds_x \;=\; 0 \quad \text{for all } v \in H_0^1(\Omega) \tag{2.3}$$

due to $-\Delta\hat{u}(x) = 0$ for $x \in \Omega_i$, where n_i denotes the exterior normal vector of Γ_i. For $x \in \Omega_i$ the solution of the local Laplace equation is given by the local representation formula

$$\hat{u}(x) \;=\; \int\limits_{\Gamma_i} U^*(x,y)\frac{\partial}{\partial n_i}\hat{u}(x)ds_x - \int\limits_{\Gamma_i} \frac{\partial}{\partial n_i}U^*(x,y)\hat{u}(y)ds_y, \tag{2.4}$$

where

$$U^*(x,y) \;=\; \frac{1}{4\pi}\frac{1}{|x-y|}$$

is the fundamental solution of the Laplace operator, see, e.g., [55]. Taking the trace and the normal derivative of the local representation formula (2.4) we obtain the Calderon system

$$\begin{pmatrix} u_i \\ t_i \end{pmatrix} = \begin{pmatrix} \frac{1}{2}I - K_i & V_i \\ D_i & \frac{1}{2}I + K_i' \end{pmatrix} \begin{pmatrix} u_i \\ t_i \end{pmatrix} \tag{2.5}$$

of local boundary integral equations on Γ_i for the Cauchy data $u_i = \hat{u}_{|\Gamma_i} \in H^{1/2}(\Gamma_i)$ and $t_i = n_i \cdot \nabla \hat{u}_{|\Gamma_i} \in H^{-1/2}(\Gamma_i)$, where the local single layer potential operator $V_i : H^{-1/2}(\Gamma_i) \to H^{1/2}(\Gamma_i)$, the local double layer potential operator $K_i : H^{1/2}(\Gamma_i) \to H^{1/2}(\Gamma_i)$, its adjoint $K_i' : H^{-1/2}(\Gamma_i) \to H^{-1/2}(\Gamma_i)$, and the local hypersingular boundary integral operator $D_i : H^{1/2}(\Gamma_i) \to H^{-1/2}(\Gamma_i)$ are defined by

$$\begin{aligned} (V_i t_i)(x) &= \int_{\Gamma_i} U^*(x,y) t_i(y) ds_y, \\ (K_i u_i)(x) &= \int_{\Gamma_i} \frac{\partial}{\partial n_{i,y}} U^*(x,y) u_i(y) ds_y, \end{aligned}$$

$$\begin{aligned} (K_i' t_i)(x) &= \int_{\Gamma_i} \frac{\partial}{\partial n_{i,x}} U^*(x,y) t_i(y) ds_y, \\ (D_i u_i)(x) &= -\frac{\partial}{\partial n_{i,x}} \int_{\Gamma_i} \frac{\partial}{\partial n_{i,y}} U^*(x,y) u_i(y) ds_y, \end{aligned}$$

respectively. The mapping properties of these boundary integral operators are now well known, see, e.g., [11, 42, 55]. In particular, the local single layer potential operators V_i are self–adjoint and $H^{-1/2}(\Gamma_i)$–elliptic. The local hypersingular operators D_i are self–adjoint as well, but only positive semi–definite. Indeed, the kernel ker D_i of the hypersingular operator D_i is spanned by all constant functions, i.e. ker $D_i = \text{span}\{1\}$.
Let us denote the skeleton of the domain decomposition by $\Gamma_S = \cup_{i=1}^p \Gamma_i$ and the skeleton trace space by $H^{1/2}(\Gamma_S) = \{v_{|\Gamma_S} : v \in H^1(\Omega)\}$. Inserting the second boundary integral equation of (2.5) into (2.3) we have to find $\hat{u} \in H^{1/2}(\Gamma_S)$ with $\hat{u}_{|\Gamma} = g$ such that

$$\sum_{i=1}^p \alpha_i \left[\langle D_i \hat{u}_{|\Gamma_i}, v_{|\Gamma_i} \rangle_{\Gamma_i} + \langle (\frac{1}{2}I + K_i') t_i, v_{|\Gamma_i} \rangle_{\Gamma_i} \right] = 0 \tag{2.6}$$

for all test functions $v \in H_0^{1/2}(\Gamma_S) = \{v \in H^{1/2}(\Gamma_S) : v_{|\Gamma} = 0\}$, where $t_i \in H^{-1/2}(\Gamma_i)$ are the unique solutions of the local variational problems

$$\alpha_i \left[\langle V_i t_i, w_i \rangle_{\Gamma_i} - \langle (\frac{1}{2}I + K_i) \hat{u}_{|\Gamma_i}, w_i \rangle_{\Gamma_i} \right] = 0 \tag{2.7}$$

for all test functions $w_i \in H^{-1/2}(\Gamma_i)$, where $\langle \cdot, \cdot \rangle_{\Gamma_i}$ denotes the duality pairing in $H^{-1/2}(\Gamma_i) \times H^{1/2}(\Gamma_i)$ that is the extension of the scalar product in $L_2(\Gamma_i)$ for $i = 1, 2, \ldots, p$. After homogenization of the Dirichlet boundary condition via the ansatz $\hat{u} = \hat{g} + u$ with some extension $\hat{g}_{|\Gamma} = g$ and $u_{|\Gamma} = 0$, we can rewrite (2.6) and (2.7) as mixed variational problem to find $t = (t_1, t_2, \ldots, t_p) \in T = T_1 \times T_2 \times \ldots \times T_p = H^{-1/2}(\Gamma_1) \times H^{-1/2}(\Gamma_2) \times \ldots \times H^{-1/2}(\Gamma_p)$ and $u \in U = H_0^{1/2}(\Gamma_S)$ such that

$$\alpha_i \left[\langle w_i, V_i t_i \rangle_{\Gamma_i} - \langle w_i, (\frac{1}{2}I + K_i) u_{|\Gamma_i} \rangle_{\Gamma_i} \right] = \alpha_i \langle w_i, (\frac{1}{2}I + K_i) \hat{g}_{|\Gamma_i} \rangle_{\Gamma_i} \tag{2.8}$$

for all $w_i \in T_i$, $i = 1, 2, \ldots, p$, and

$$\sum_{i=1}^{p} \alpha_i \left[-\langle (\frac{1}{2} I + K_i') t_i, v_{|\Gamma_i} \rangle_{\Gamma_i} - \langle D_i u_{|\Gamma_i}, v_{|\Gamma_i} \rangle_{\Gamma_i} \right] \; = \; \sum_{i=1}^{p} \alpha_i \langle D_i \hat{g}_{|\Gamma_i}, v_{|\Gamma_i} \rangle_{\Gamma_i} \tag{2.9}$$

for all $v \in U$. It is well known that the mixed variational problem (2.8) and (2.9) is uniquely solvable, and therefore the equivalent problem (2.6) and (2.7), too [31].

Let us now introduce the boundary element trial spaces

$$U_h = S_h^1(\Gamma_S) = \operatorname{span}\{\varphi_m\}_{m=1}^{M} \subset U \; = \; H_0^{1/2}(\Gamma_S) \tag{2.10}$$

and

$$T_{i,h} = S_h^0(\Gamma_i) = \operatorname{span}\{\psi_k^i\}_{k=1}^{N_i} \subset T_i \; = \; H^{-1/2}(\Gamma_i) \tag{2.11}$$

spanned by continuous piecewise linear basis functions φ_m and by piecewise constant basis functions ψ_k^i with respect to a regular globally quasi–uniform boundary element mesh $\Gamma_i = \cup_{\ell=1}^{N_i} \tau_\ell^i$ with the average mesh size h on Γ_S and Γ_i, respectively. The restriction of the global trial space U_h onto the local subdomain boundaries Γ_i is denoted by

$$U_{i,h} = S_h^1(\Gamma_i) = S_h^1(\Gamma_S)_{|\Gamma_i} = \operatorname{span}\{\varphi_m^i\}_{m=1}^{M_i} \subset H_0^{1/2}(\Gamma_i), \tag{2.12}$$

where $H_0^{1/2}(\Gamma_i) = \{v_{|\Gamma_i} : v \in H_0^{1/2}(\Gamma_S)\}$. The Galerkin discretization of the symmetric mixed variational formulation (2.8) and (2.9) now gives immediately the mixed boundary element equations to find $t_h = (t_{1,h}, t_{2,h}, \ldots, t_{p,h}) \in T_h = T_{1,h} \times T_{2,h} \times \ldots \times T_{p,h}$ and $u_h \in U_h$ such that

$$\alpha_i \left[\langle w_{i,h}, V_i t_{i,h} \rangle_{\Gamma_i} - \langle w_{i,h}, (\frac{1}{2} I + K_i) u_{h|\Gamma_i} \rangle_{\Gamma_i} \right] \; = \; \alpha_i \langle w_{i,h}, (\frac{1}{2} I + K_i) \hat{g}_{|\Gamma_i} \rangle \tag{2.13}$$

for all $w_{i,h} \in T_{i,h}$, $i = 1, 2, \ldots, p$, and

$$\sum_{i=1}^{p} \alpha_i \left[-\langle (\frac{1}{2} I + K_i') t_{i,h}, v_{h|\Gamma_i} \rangle_{\Gamma_i} - \langle D_i u_{h|\Gamma_i}, v_{h|\Gamma_i} \rangle_{\Gamma_i} \right] \; = \; \sum_{i=1}^{p} \alpha_i \langle D_i \hat{g}_{|\Gamma_i}, v_{h|\Gamma_i} \rangle_{\Gamma_i} \tag{2.14}$$

for all $v_h \in U_h$. Applying standard arguments, it follows that the mixed boundary element equations (2.13) and (2.14) admit an unique solution. Moreover, the discretization error is bounded by the approximation error,

$$\|u - u_h\|_U^2 + \|t - t_h\|_T^2 \le c \left\{ \inf_{v_h \in U_h} \|u - v_h\|_U^2 + \inf_{w_h \in T_h} \|t - w_h\|_T^2 \right\}.$$

From the approximation properties of the boundary element trial spaces U_h and T_h we then obtain the a priori error estimate

$$\|u - u_h\|_U^2 + \|t - t_h\|_T^2 \; \le \; c\, h^{2s+1} \left[\|u\|_{H^{s+1}(\Gamma_S)}^2 + \sum_{i=1}^{p} \|t_i\|_{H_{\text{pw}}^s(\Gamma_i)}^2 \right] \tag{2.15}$$

for a sufficiently smooth solution (u, t) with $u \in H^{s+1}(\Gamma_S)$ and $t_i \in H_{\text{pw}}^s(\Gamma_i)$ and $s \in (0, 1]$. This gives an optimal convergence rate of $\mathcal{O}(h^{3/2})$, see [55]. Here and in the following c denotes a generic constant that is always independent on the discretization parameters.

Once the basis functions are chosen, the boundary element equations (2.13) and (2.14) are equivalent to the following system of algebraic equations to find the coefficient vectors $\underline{t}_i = [t_{i,k}]_{k=1}^{N_i} \in I\!R^{N_i}$ for $i = 1, 2, \ldots, p$ and $\underline{u} = [u_m]_{m=1}^{M} \in I\!R^M$ of the boundary element functions $t_{i,h} = \sum_{k=1}^{N_i} t_{i,k} \psi_k^i$ and $u_h = \sum_{m=1}^{M} u_m \varphi_m$ as the unique solution of the Galerkin system

$$\begin{pmatrix} \alpha_1 V_{1,h} & & & -\alpha_1 \bar{K}_{1,h} R_{1,h} \\ & \ddots & & \vdots \\ & & \alpha_p V_{p,h} & -\alpha_p \bar{K}_{p,h} R_{p,h} \\ -\alpha_1 R_{1,h}^\top \bar{K}_{1,h}^\top & \ldots & -\alpha_p R_{p,h}^\top \bar{K}_{p,h}^\top & -D_h \end{pmatrix} \begin{pmatrix} \underline{t}_1 \\ \vdots \\ \underline{t}_p \\ \underline{u} \end{pmatrix} = \begin{pmatrix} \alpha_1 \underline{g}_1 \\ \vdots \\ \alpha_p \underline{g}_p \\ \underline{f} \end{pmatrix}, \tag{2.16}$$

where

$$D_h = \sum_{i=1}^{p} \alpha_i R_{i,h}^\top D_{i,h} R_{i,h} \quad \text{and} \quad \underline{f} = \sum_{i=1}^{p} \alpha_i R_{i,h}^\top \underline{f}_i. \tag{2.17}$$

The entries of the matrices $V_{i,h}$, $\bar{K}_{i,h}$, $D_{i,h}$ and the coefficients of the right–hand sides $\underline{g}_i$ and $\underline{f}$ are defined by the relations

$$\begin{aligned} V_{i,h}[\ell,k] &= \langle \psi_\ell^i, V_i \psi_k^i \rangle_{\Gamma_i}, \\ \bar{K}_{i,h}[\ell,m] &= \langle \psi_\ell^i, (\frac{1}{2}I + K_i)\varphi_m^i \rangle_{\Gamma_i}, \\ D_{i,h}[n,m] &= \langle \varphi_n^i, D_i \varphi_m^i \rangle_{\Gamma_i} \end{aligned}$$

and

$$g_{i,\ell} = \langle \psi_\ell^i, (\frac{1}{2}I + K_i)\hat{g}_{|\Gamma_i} \rangle_{\Gamma_i}, \quad f_{i,n} = \langle D_i \hat{g}_{|\Gamma_i}, \varphi_n^i \rangle_{\Gamma_i}$$

for all $k, \ell = 1, \ldots, N_i$ and $m, n = 1, \ldots, M_i$, respectively. The restriction operator $R_{i,h}$ maps some global coefficient vector $\underline{v} \in I\!R^M$ to the local vector $\underline{v}_i \in I\!R^{M_i}$ containing those components of $\underline{v}$ which correspond to Γ_i only, $i = 1, 2, \ldots, p$. The matrices $R_{i,h}$ are Boolean matrices which are sometimes also called subdomain connectivity matrices.

The system matrix of (2.16) is symmetric, but indefinite. The matrices $V_{i,h}$ are symmetric and positive definite while the blocks $D_{i,h}$ are symmetric, but in general only positive semi–definite, where the kernel is spanned by all vectors with constant entries. Unfortunately, all matrix blocks involved in (2.16) are fully populated due to the nonlocal fundamental solution. Especially for three–dimensional boundary value problems, this fact would yield unacceptable high resources of memory and already a high complexity of a single matrix–by–vector multiplication. Since the latter operation is the basic operation in every iteration method, we need some data–sparse approximation of these matrices that reduces the quadratic complexity with respect to the number of unknowns to an almost linear one, but without disturbing the accuracy given by the quasi–optimal estimate (2.15).

As already pointed out in the introduction there exist several possibilities of fast boundary element methods to reduce the computational costs, for example panel clustering [28], algebraic approximation techniques such as the Adaptive Cross Approximation method [3] and hierarchical matrices [27] or wavelets [12]. Here we describe the application of the fast multipole method to all the discrete boundary integral operators introduced above [46]. We will mainly use the original version of the fast multipole method as described in [21, 22] but the main difference in our version of the method, however, will be the use of reformulated spherical harmonics [48, 59, 60].

As already mentioned above, when using an iterative solver it is necessary to perform the matrix–by–vector multiplications of the local stiffness matrices with given vectors efficiently. In particular, if the vectors $\underline{r}_1^i \in I\!R^{N_i}$ and $\underline{r}_2^i \in I\!R^{M_i}$ are given, we need a fast computation of the matrix–by–vector products $\underline{w}_1^i = V_{i,h}\underline{r}_1^i \in I\!R^{N_i}$, $\underline{w}_2^i = K_{i,h}\underline{r}_2^i \in I\!R^{N_i}$, $\underline{v}_1^i = D_{i,h}\underline{r}_2^i \in I\!R^{M_i}$ and of $\underline{v}_2^i = K_{i,h}^\top \underline{r}_1^i \in I\!R^{M_i}$.

We begin with the local matrix–by–vector product $\underline{w}_1^i = V_{i,h}\underline{r}_1^i$ of the discrete single layer potential $V_{i,h}$ which reads in components

$$w_{1,\ell}^i = \sum_{k=1}^{N_i} V_{i,h}[\ell,k] r_{1,k}^i = \sum_{k=1}^{N_i} \frac{r_{1,k}^i}{4\pi} \int_{\tau_\ell^i} \int_{\tau_k^i} \frac{1}{|x-y|} ds_y ds_x \quad \text{for } \ell = 1, \ldots, N_i.$$

As in most of the fast boundary element methods, one first substitutes the kernel function by an appropriate series expansion separating the integration variables x and y from each other. The kernel of the single layer potential can be written in terms of Legendre polynomials,

$$k(x,y) = \frac{1}{|x-y|} = \sum_{n=0}^{\infty} \frac{|x|^n}{|y|^{n+1}} P_n(\hat{x} \cdot \hat{y}), \quad \hat{x} = \frac{x}{|x|}, \quad \hat{y} = \frac{y}{|y|},$$

where

$$P_n(u) \;=\; \frac{1}{2^n n!}\frac{d^n}{du^n}(u^2-1)^n \quad \text{for } |u| \le 1.$$

An approximation of the kernel function $k(x, y)$ is then defined by truncating the infinite sum at at certain expansion degree ϱ,

$$k_\varrho(x,y) \;=\; \sum_{n=0}^{\varrho} \frac{|x|^n}{|y|^{n+1}} P_n(\hat{x}\cdot\hat{y}). \tag{2.18}$$

Note that for $|y| > |x|$ the following error estimate holds:

$$|k(x,y) - k_\varrho(x,y)| \;\le\; \frac{1}{|y|-|x|}\left(\frac{|x|}{|y|}\right)^{\varrho+1}. \tag{2.19}$$

Since the product of the normalized integration variables $\hat{x}$ and $\hat{y}$ still appears as an argument of the Legendre polynomials in (2.18), one may rather use spherical harmonics [29] to obtain the kernel approximation

$$k_\varrho(x,y) \;=\; \sum_{n=0}^{\varrho}\sum_{m=-n}^{n} \frac{|x|^n}{|y|^{n+1}} Y_n^{-m}(\hat{x})Y_n^m(\hat{y}) \;=\; \sum_{n=0}^{\varrho}\sum_{m=-n}^{n} \overline{S_n^m}(y)R_n^m(x) \tag{2.20}$$

when assuming $|y| > |x|$. The truncated series expansion (2.20) is the first ingredient of the fast multipole method. The second one is the use of an artificial geometric hierarchy which is build upon the boundary elements. All boundary elements $\tau_k^i \subset \Gamma_i$ belong to a cluster $\omega_{i,1}^0$ which is a box containing the whole subdomain Ω_i. The clusters of the level $\lambda+1$ are constructed by the refinement of all boxes $\omega_{i,j}^\lambda$ into eight similar sub–boxes $\omega_{i,j'}^{\lambda+1}$. The clusters $\omega_{i,j'}^{\lambda+1}$ are called the sons of the father cluster $\omega_{i,j}^\lambda$. This recursive process is repeated until an maximum level L is attained. Clusters containing no boundary elements are neglected. All boundary elements $\{\tau_k^i\}_{k=1}^{N_i}$ are assigned to the cluster $\omega_{i,j}^L$ with respect to their midpoints.
Since the expansion (2.20) is only valid for $|x| < |y|$, one has to distinguish the farfield (FF) where this expansion is admissible from the nearfield (NF) where the standard kernel has to be used. A cluster ω_j^λ on the same level λ is called to be in the nearfield of the cluster ω_i^λ if the condition

$$\operatorname{dist}\{C_i^\lambda, C_j^\lambda\} \;\le\; (d+1)\,\max\{r_i^\lambda, r_j^\lambda\} \tag{2.21}$$

is satisfied with $d > 1$ a suitably chosen parameter. C_i^λ denotes the center and r_i^λ is the radius of the cluster ω_i^λ which is determined by the cluster center and the extension of the contained boundary elements $\tau_k \in \omega_i^\lambda$.
Hence we obtain the approximate local matrix–by–vector product

$$\begin{aligned}
\widetilde{w}_{1,\ell}^i \;&=\; \sum_{k\,\in \mathrm{NF}(\ell)} V_{i,h}[\ell,k] r_{1,k}^i + \sum_{k\,\in \mathrm{FF}(\ell)} \frac{r_{1,k}^i}{4\pi}\int_{\tau_\ell}\int_{\tau_k}\sum_{n=0}^{\varrho}\sum_{m=-n}^{n}\overline{S_n^m}(y)R_n^m(x)ds_x ds_y \\
&=\; \sum_{k\,\in \mathrm{NF}(\ell)} V_{i,h}[\ell,k] r_{1,k}^i + \sum_{n=0}^{\varrho}\sum_{m=-n}^{n}\sum_{k\,\in \mathrm{FF}(\ell)} \frac{r_{1,k}^i}{4\pi}\int_{\tau_k} R_n^m(x)ds_x \int_{\tau_\ell}\overline{S_n^m}(y)ds_y
\end{aligned}$$

for all $\ell = 1, \ldots, N_i$. As the farfields differ from each other for most indices ℓ, the question arises how to compute the coefficients of the series expansion for different ℓ efficiently. This can be done by using the cluster hierarchy constructed above. Starting on the finest level all coefficients for the corresponding clusters are computed and later translated to their common fathers on the coarser levels up to level zero. Next, these multipole expansions are converted into local expansions in terms of their coefficients on the highest level, in which the kernel expansion is admissible. This results in an algorithm of $\mathcal{O}(N_i \log^2 N_i)$ complexity. For a more detailed description of the efficient matrix–by–vector multiplication see [46] and the references given there.
In what follows we will focus on the stability and error analysis of the approximate matrix–by–vector product $\underline{\widetilde{w}}_1^i = \widetilde{V}_{i,h}\underline{r}_1^i$ in the case of a fixed nearfield parameter $d > 0$.

Lemma 2.1 [46, Lemma 4.1] *Let $\varrho = \mathcal{O}(\log h^{-2})$ and the triangulation be quasi–uniform. Then there holds the following consistency estimate for the approximated single layer potential $\widetilde{V}_i$,*

$$\|(V_i - \widetilde{V}_i)t_{i,h}\|_{L_2(\Gamma_i)} \le c\,h^2 \|t_{i,h}\|_{L_2(\Gamma_i)} \quad \textit{for all} \quad t_{i,h} \in S_h^0(\Gamma_i) \subset L_2(\Gamma_i).$$

From the previous lemma and using the inverse inequality it is possible to choose the expansion degree $\varrho = \mathcal{O}(\log N_i)$ such that the approximate single layer potential $\widetilde{V}_{i,h}$ is $S_h^0(\Gamma_i)$–elliptic,

$$\langle \widetilde{V}_{i,h} w_{i,h}, w_{i,h}\rangle_{\Gamma_i} \ge c_1^{\widetilde{V}_i} \|w_{i,h}\|^2_{H^{-1/2}(\Gamma)} \quad \text{for all } w_{i,h} \in S_h^0(\Gamma_i).$$

The multipole expansion of the double layer potential operator $\widetilde{K}_i$ can be computed by applying the normal derivative to the expansion (2.20) of the single layer potential V_i. In the case of the adjoint double layer potential K_i', the normal derivative has to be applied to the coefficients in the evaluation of the expansions. Therefore the transposition property between the Galerkin matrices of the the double layer potential and its adjoint operator will be maintained if the same expansion degree p is chosen for both. As for the approximate single layer potential $\widetilde{V}_{i,h}$ corresponding error estimates follow, see [46, Lemma 4.3, Lemma 4.4].

The hypersingular boundary integral operator D_i is treated via integration by parts [44]. The bilinear form of the hypersingular integral operator D_i can be written as sum of bilinear forms of the single layer potential V_i,

$$\langle D_i u_i, v_i\rangle_{\Gamma_i} = \sum_{k=1}^{3} \langle V_i \underline{\mathrm{curl}}_{|\Gamma_i} u_{i|k}, \underline{\mathrm{curl}}_{|\Gamma_i} v_{i|k}\rangle_{\Gamma_i},$$

with the surface curl $\underline{\mathrm{curl}}_{|\Gamma_i} u_i(x) = n_x \times \nabla u^*_{i,\Gamma_i}(x)$ defined for a constant extension $u^*_{i|\Gamma_i}$ of $u(x)$ along the normal n_x into a small neighborhood of Γ_i. In the case of piecewise linear, continuous basis functions $\varphi^i_m(x)$ and plane triangles as boundary elements, the vector $\underline{\mathrm{curl}}_{\Gamma_i}\varphi^i_m(x)$ is piecewise constant and constant on each boundary element. Therefore the Galerkin matrix of the single layer potential V_i for piecewise constant basis functions ψ^i_k can be reused here leading to the approximate bilinear form

$$\langle \widetilde{D}_{i,h} u_{i,h}, v_{i,h}\rangle_{\Gamma_i} = \sum_{k=1}^{3} \langle \widetilde{V}_{i,h} \underline{\mathrm{curl}}_{|\Gamma_i} u_{i,h|k}, \underline{\mathrm{curl}}_{|\Gamma_i} v_{i,h|k}\rangle_{\Gamma_i}. \tag{2.22}$$

The stability and error analysis for the approximation $\widetilde{D}_{i,h}$ then follows as for the approximate single layer potential.

Using the fast multipole approximations for the application of the discrete boundary integral operators, instead of (2.16) we have to solve the perturbed linear system

$$\begin{pmatrix} \alpha_1 \widetilde{V}_{1,h} & & & -\alpha_1 \widetilde{K}_{1,h} R_{1,h} \\ & \ddots & & \vdots \\ & & \alpha_p \widetilde{V}_{p,h} & -\alpha_p \widetilde{K}_{p,h} R_{p,h} \\ -\alpha_1 R_{1,h}^\top \widetilde{K}_{1,h}^\top & \dots & -\alpha_p R_{p,h}^\top \widetilde{K}_{p,h}^\top & -\widetilde{D}_h \end{pmatrix} \begin{pmatrix} \underline{\widetilde{t}}_1 \\ \vdots \\ \underline{\widetilde{t}}_p \\ \underline{\widetilde{u}} \end{pmatrix} = \begin{pmatrix} \alpha_1 \underline{\widetilde{g}}_1 \\ \vdots \\ \alpha_p \underline{\widetilde{g}}_p \\ \underline{\widetilde{f}} \end{pmatrix}. \tag{2.23}$$

As for the symmetric boundary integral formulation considered in [46, Theorem 4.1] there follows the unique solvability of (2.23) as well as the quasi–optimal error estimate

$$\|u - \widetilde{u}_h\|_U^2 + \|t - \widetilde{t}_h\|_T^2 \le c\,h^{2s+1} \left[\|u\|^2_{H^{s+1}(\Gamma_S)} + \sum_{i=1}^{p} \|t_i\|^2_{H^s_{\mathrm{pw}}(\Gamma_i)} \right] \tag{2.24}$$

for the associated approximate solutions $\widetilde{t}_h \in T_h$ and $\widetilde{u}_h \in U_h$ when assuming $\varrho = \mathcal{O}(\log N_i)$ and smooth solutions as in (2.15), i.e., $s \in (0, 1]$. Hence we can ensure the same asymptotic behavior as for the solution of the standard Galerkin boundary element method.

A standard approach to solve the linear system (2.23) is to consider its Schur complement system

$$\sum_{i=1}^{p} \alpha_i R_{i,h}^{\top} \left[\widetilde{D}_{i,h} + \widetilde{K}_{i,h}^{\top} \widetilde{V}_{i,h}^{-1} \widetilde{K}_{i,h} \right] R_{i,h} \underline{u} \;=\; -\sum_{i=1}^{p} \alpha_i R_{i,h}^{\top} \left[\underline{f}_i + \widetilde{K}_{i,h}^{\top} \widetilde{V}_{i,h}^{-1} \underline{g}_i \right] \tag{2.25}$$

with the symmetric and positive definite system matrix

$$\widetilde{S}_h \;=\; \sum_{i=1}^{p} \alpha_i R_{i,h}^{\top} \widetilde{S}_{i,h} R_{i,h}, \quad \widetilde{S}_{i,h} = \widetilde{D}_{i,h} + \widetilde{K}_{i,h}^{\top} \widetilde{V}_{i,h}^{-1} \widetilde{K}_{i,h}.$$

The linear system (2.25) can be solved by a conjugate gradient scheme where the application of a possible preconditioner is given by

$$C_S^{-1} \;=\; \sum_{i=1}^{p} \frac{1}{\alpha_i} R_{i,h}^{\top} \bar{V}_{i,h} R_{i,h} \tag{2.26}$$

and $\bar{V}_{i,h}$ is the single layer potential discretized by piecewise linear continuous basis functions and approximated by means of a fast multipole method. Note that each application of the discrete Steklov–Poincaré operators $\widetilde{S}_{i,h}$ within one step of the conjugate gradient scheme requires one inversion of the discrete single layer potential $\widetilde{V}_{i,h}$ which can be realized by an interior conjugate gradient scheme.

3 Inexact BETI Methods

In order to avoid assembled matrices and vectors, we tear the global potential vector $\underline{u}$ on the subdomain boundaries Γ_i by introducing the individual local unknowns

$$\underline{u}_i = R_{i,h} \underline{u}. \tag{3.1}$$

Conversely, the global continuity of the potentials is enforced by the constraints

$$\sum_{i=1}^{p} B_i \underline{u}_i = \underline{0} \tag{3.2}$$

interconnecting the local potential vectors across the subdomain boundaries. Each row of the matrix $B = (B_1, \ldots, B_p)$ is connected with a pair of matching nodes across the subdomain boundaries. The entries of such a row are 1 and -1 for the indices corresponding to the matching nodes on the interface (coupling boundaries) $\Gamma_C = \Gamma_S \setminus \Gamma$ and 0 otherwise. Therefore, (3.2) implies that the corresponding boundary element functions $u_{i,h}$ are continuous across the interface Γ_C, i.e. $u_{i,h} = u_{j,h}$ on $\Gamma_i \cap \Gamma_j \neq \emptyset$. We assume here that the number of constraints at some matching node is equal to the number of matching subdomains minus one. This method of a minimal number of constraints respectively multipliers is called non–redundant, see, e.g., [34] for the use of redundant constraints.
By introducing Lagrange multipliers $\underline{\lambda} \in I\!R^L$ the linear system (2.23) is equivalent to

$$\begin{pmatrix} \alpha_1 \widetilde{V}_{1,h} & & & -\alpha_1 \widetilde{K}_{1,h} & & & \\ & \ddots & & & \ddots & & \\ & & \alpha_p \widetilde{V}_{p,h} & & & -\alpha_p \widetilde{K}_{p,h} & \\ -\alpha_1 \widetilde{K}_{1,h}^{\top} & & & -\alpha_1 \widetilde{D}_{1,h}^{\top} & & & B_1^{\top} \\ & \ddots & & & \ddots & & \vdots \\ & & -\alpha_p \widetilde{K}_{p,h}^{\top} & & & -\alpha_p \widetilde{D}_{i,h} & B_p^{\top} \\ & & & B_1 & \ldots & B_p & \end{pmatrix} \begin{pmatrix} \underline{\widetilde{t}}_1 \\ \vdots \\ \underline{\widetilde{t}}_p \\ \underline{\widetilde{u}}_1 \\ \vdots \\ \underline{\widetilde{u}}_p \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} \alpha_1 \underline{g}_1 \\ \vdots \\ \alpha_p \underline{g}_p \\ \alpha_1 \underline{f}_1 \\ \vdots \\ \alpha_p \underline{f}_p \\ \underline{0} \end{pmatrix}.$$

Setting $V = \mathrm{diag}(\alpha_i \widetilde{V}_{i,h})$, $K = \mathrm{diag}(\alpha_i \widetilde{K}_{i,h})$, $D = \mathrm{diag}(\alpha_i \widetilde{D}_{i,h})$ as well as $B = (B_1, \ldots, B_p)$ we have to solve the two–fold saddle point problem

$$\begin{pmatrix} V & -K & 0 \\ -K^\top & -D & B^\top \\ 0 & B & 0 \end{pmatrix} \begin{pmatrix} \underline{t} \\ \underline{u} \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} \underline{g} \\ \underline{f} \\ \underline{0} \end{pmatrix} \tag{3.3}$$

when ordering the vectors in an appropriate way. The matrix $\widetilde{D}_{i,h}$ is symmetric and positive definite if and only if the corresponding subdomain Ω_i is a so–called non–floating subdomain. A subdomain Ω_i is called floating if the subdomain boundary Γ_i does not touch the Dirichlet boundary part Γ_D that is Γ for our model problem (2.1). Otherwise the subdomain is called non–floating. For a floating subdomain Ω_i the kernel of the matrix $\widetilde{D}_{i,h}$ is spanned by the vector $\underline{e}_i = (1, \ldots, 1)^\top \in I\!R^{M_i}$, i.e. $\ker \widetilde{D}_{i,h} = \mathrm{span}\{\underline{e}_i\}$. Let us assume for simplicity that the first q ($q < p$) subdomains are floating and the remaining subdomains are non–floating. The case $q = 0$ (no floating subdomains) is trivial and does not need any discussion. In order to avoid singular block entries $\widetilde{D}_{i,h}$ and, therefore, singular Schur complements $\widetilde{S}_{i,h} = \alpha_i(\widetilde{D}_{i,h} + \widetilde{K}_{i,h}^\top \widetilde{V}_{i,h}^{-1} \widetilde{K}_{i,h})$ for the floating subdomains, we replace the singular blocks $\widetilde{D}_{i,h}$ by the regularized matrices $\widetilde{D}_{i,h} + \beta_i \underline{e}_i \underline{e}_i^\top$ with appropriately choosen positive constants β_i for $i = 1, \ldots, q$. We mention that

$$\widetilde{K}_{i,h} \underline{e}_i = \underline{0} \quad \text{for } i = 1, \ldots, q. \tag{3.4}$$

Let us denote the unique solutions of the regularized equations

$$\alpha_i(\widetilde{D}_{i,h} + \beta_i \underline{e}_i \underline{e}_i^\top) \underline{v}_i \;=\; -\alpha_i \underline{f}_i + B_i \underline{\lambda} - \alpha_i \widetilde{K}_{i,h}^\top \underline{t}_i \tag{3.5}$$

by $\underline{v}_i$, $i = 1, \ldots, q$. If the right hand side of (3.5) fulfills the solvability conditions for the original equations without the regularization term,

$$\underline{e}_i^\top(-\alpha_i \underline{f}_i + B_i \underline{\lambda} - \alpha_i \widetilde{K}_{i,h}^\top \underline{t}_i) \;=\; \underline{e}_i^\top(-\alpha_i \underline{f}_i + B_i \underline{\lambda}) \;=\; 0, \tag{3.6}$$

then the solution $\underline{v}_i$ of (3.5) is orthogonal to $\ker \widetilde{D}_{i,h}$, i.e.

$$\underline{e}_i^\top \underline{v}_i = 0. \tag{3.7}$$

Now the solution $\underline{u} = (\underline{u}_1, \ldots, \underline{u}_p)^\top$ of the original system (3.3) can be recovered by the formulae

$$\underline{u}_i = \underline{v}_i + \gamma_i \underline{e}_i, \tag{3.8}$$

with $\gamma_i = 0$ for $i = q+1, \ldots, p$ and appropriately chosen γ_i for $i = 1, \ldots, q$. Taking into account (3.4)–(3.8), we easily observe that the unique solution $\underline{t} \in I\!R^M$, $\underline{v} \in I\!R^N$, $\underline{\lambda} \in I\!R^L$ and $\underline{\gamma} \in I\!R^q$ of the three–fold saddle point problem

$$\begin{pmatrix} V & -K & 0 & 0 \\ -K^\top & -D & B^\top & 0 \\ 0 & B & 0 & G \\ 0 & 0 & G^\top & 0 \end{pmatrix} \begin{pmatrix} \underline{t} \\ \underline{v} \\ \underline{\lambda} \\ \underline{\gamma} \end{pmatrix} = \begin{pmatrix} \underline{g} \\ \underline{f} \\ \underline{0} \\ \underline{e} \end{pmatrix} \tag{3.9}$$

immediately yields the solution $(\underline{t}, \underline{u}, \underline{\lambda})$ of the two–fold saddle point problem (3.3), where now D is symmetric and positive definite since we replaced the singular blocks $\widetilde{D}_{j,h}$ by the regularized ones for the floating subdomains. The last equation in (3.9) is nothing but the solvability condition (3.6). Note that the $L \times q$ matrix G and the vector $\underline{e} \in I\!R^q$ are defined by the relations

$$G = (B_1 \underline{e}_1, \ldots, B_q \underline{e}_1) \quad \text{and} \quad \underline{e} = (\underline{e}_1^\top \underline{f}_1, \ldots, \underline{e}_q^\top \underline{f}_q)^\top. \tag{3.10}$$

Now we use a subspace projection in order to separate the determination of $\underline{\gamma}$ from the determination of the rest of the unknowns in (3.9). Thus, we introduce the orthogonal projection

$$P = I - G(G^\top G)^{-1} G^\top \tag{3.11}$$

from the space $\mathbf{\Lambda} := I\!R^M$ onto the subspace $\mathbf{\Lambda}_0 = \ker G^\top = (\mathrm{range}\, G)^\perp$ with respect to the scalar product $(\cdot,\cdot) = (\cdot,\cdot)_{\mathbf{\Lambda}} = (\cdot,\cdot)_{I\!R^M}$. We mention that in the case of large jumps in the coefficients of the PDE the scalar product in $\mathbf{\Lambda}$ has to be changed according to the proposal made in [34], see also [7] and [58]. Of course, the change of the scalar product changes the orthoprojection P too. Since $PG = 0$, the application of P to the third equation of (3.9) gives $PB\underline{v} = 0$ that excludes $\underline{\gamma}$ from the first three equations of (3.9). Let us represent $\underline{\lambda}$ in the form

$$\underline{\lambda} = T_0\underline{\lambda}_0 + \underline{\lambda}_e \tag{3.12}$$

with known $\underline{\lambda}_e = G(G^\top G)^{-1}\underline{e} \in (\ker G^\top)^\perp = \mathrm{range}\, G$, fulfilling the constraints $G\top\underline{\lambda}_e = \underline{e}$, and unknown $T_0\underline{\lambda}_0 \in \ker G^\top$, i.e. $G^\top\underline{\lambda}_0 = 0$, where $\underline{\lambda}_0 \in I\!R^{L_0}$ and $L_0 = \dim \Lambda_0$. The columns of T_0 span a basis of Λ_0. We refer to Remark 3.1 for the practical implementation of the determination of $\underline{\lambda}_0$ via subspace iteration. Now we can define $\underline{t}, \underline{v}$ and $\underline{\lambda}_0$ from the two–fold saddle point problem

$$\begin{pmatrix} V & -K & 0 \\ -K^\top & -D & B^\top P^\top T_0 \\ 0 & T_0^\top PB & 0 \end{pmatrix} \begin{pmatrix} \underline{t} \\ \underline{v} \\ \underline{\lambda}_0 \end{pmatrix} = \begin{pmatrix} \underline{g} \\ \underline{d} \\ \underline{0} \end{pmatrix}, \tag{3.13}$$

where here $\underline{d} = \underline{f} - B^\top\underline{\lambda}_e$. Once the vectors $\underline{t}$, $\underline{v}$ and $\underline{\lambda}_0$ are defined from (3.13), we get $\underline{\lambda}$ from (3.12), $\underline{\gamma}$ from the third equation of (3.9), i.e.

$$\underline{\gamma} = -(G^\top G)^{-1}G^\top B\underline{v}, \tag{3.14}$$

and, finally, $\underline{u}$ from (3.8).

Remark 3.1 *We mention that the iteration updates for $\underline{\lambda}$ completely life in the subspace $\mathbf{\Lambda}_0$ if the initial guess is chosen as $\underline{\lambda}_e$. Therefore, a basis of $\mathbf{\Lambda}_0$, forming the columns of T_0, is not explicitly needed in the computation (see also Algorithm 5.1 in Section 5).*

Since $V = \mathrm{diag}(\alpha_i \widetilde{V}_{i,h})$ is block diagonal and therefore easily invertible we also may eliminate the vector $\underline{t}$ in (3.13) to obtain

$$\begin{pmatrix} S & -B^\top P^\top T_0 \\ T_0^\top PB & 0 \end{pmatrix} \begin{pmatrix} \underline{v} \\ \underline{\lambda}_0 \end{pmatrix} = \begin{pmatrix} -\underline{d} - V^{-1}\underline{g} \\ \underline{0} \end{pmatrix} \tag{3.15}$$

with

$$S = \mathrm{diag}(\alpha_i \widetilde{S}_{i,h}), \quad \widetilde{S}_{i,h} = \widetilde{D}_{i,h} + \widetilde{K}_{i,h}^\top \widetilde{V}_{i,h}^{-1} \widetilde{K}_{i,h}.$$

Eliminating $\underline{v}$ we have to solve the Schur complement system of (3.15),

$$F\underline{\lambda}_0 = T_0^\top PBS^{-1}B^\top P^\top T_0\underline{\lambda}_0 = T_0^\top PBS^{-1}[\underline{d} + V^{-1}\underline{g}]. \tag{3.16}$$

Appropriate strategies to solve (3.16) were considered in [38]; another choice is the application of a Bramble–Pasciak conjugate gradient scheme to the one–fold saddle point problem (3.15). Note that appropriate preconditioners for both methods coincide with preconditioners already needed for solving (3.13), see Section 5.

4 Solution of Two–fold Saddle Point Problems

In this section we construct and analyze preconditioned CG–like iterative methods for solving two–fold saddle point problems possessing exactly the structure and the properties of the two–fold saddle point problem (3.13), namely

$$\begin{pmatrix} A_1 & B_1^\top & 0 \\ B_1 & -A_2 & B_2^\top \\ 0 & B_2 & 0 \end{pmatrix} \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{pmatrix} = \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \underline{b}_3 \end{pmatrix}, \tag{4.1}$$

where $S_1 = A_1$, A_2, $S_2 = A_2 + B_1 A_1^{-1} B_1^\top$ and $S_3 = B_2 S_2^{-1} B_2^\top$ are symmetric and positive definite. It is easy to see that the system matrix

$$\mathcal{K}_3 = \begin{pmatrix} A_1 & B_1^\top & 0 \\ B_1 & -A_2 & B_2^\top \\ 0 & B_2 & 0 \end{pmatrix}$$

of (4.1) possesses a block LU–decomposition:

$$\mathcal{K}_3 = \mathcal{L}_3 \mathcal{U}_3 \tag{4.2}$$

with

$$\mathcal{L}_3 = \begin{pmatrix} S_1 & & \\ B_1 & -S_2 & \\ & B_2 & -S_3 \end{pmatrix} \quad \text{and} \quad \mathcal{U}_3 = \begin{pmatrix} I & S_1^{-1} B_1^T & \\ & I & -S_2^{-1} B_2^T \\ & & I \end{pmatrix}$$

The factor $\mathcal{L}_3$ motivates the following construction of a preconditioner $\hat{\mathcal{L}}_3$: We start by replacing S_1, S_2 and S_3 by preconditioners $\hat{S}_1$, $\hat{S}_2$ and $\hat{S}_3$. Additionally, we introduce two scaling factors. First the left upper block is premultiplied by some factor $\tau_1 > 0$ leading to a 2–by–2 sub matrix

$$\hat{\mathcal{L}}_2 = \begin{pmatrix} \tau_1 \hat{S}_1 & \\ B_1 & -\hat{S}_2 \end{pmatrix}.$$

We end up with the final form of the preconditioner by premultiplying this sub matrix by some second scaling factor $\tau_2 > 0$:

$$\hat{\mathcal{L}}_3 = \begin{pmatrix} \tau_2 \hat{\mathcal{L}}_2 & \\ B_2 \mathsf{E}_2 & \hat{S}_3 \end{pmatrix} = \begin{pmatrix} \tau_2 \tau_1 \hat{S}_1 & & \\ \tau_2 B_1 & -\tau_2 \hat{S}_2 & \\ & B_2 & \hat{S}_3 \end{pmatrix}$$

with $\mathsf{E}_2 = \begin{pmatrix} 0 & I \end{pmatrix}$. Now we have:

Theorem 4.1 *If the parameter τ_1 is chosen such that*

$$0 < \tau_1 < \lambda_{\min}(\hat{S}_1^{-1} A_1)$$

then the matrix $\hat{\mathcal{L}}_2^{-1} \mathcal{K}_2$ is symmetric and positive definite in the scalar product $(u, v)_2 = (\mathcal{D}_2 u, v)_{\ell_2}$ with

$$\mathcal{K}_2 = \begin{pmatrix} A_1 & B_1^T \\ B_1 & -A_2 \end{pmatrix} \quad \textit{and} \quad \mathcal{D}_2 = \begin{pmatrix} A_1 - \tau_1 \hat{S}_1 & \\ & \hat{S}_2 \end{pmatrix}.$$

If, additionally, the parameter τ_2 is chosen such that

$$0 < \tau_2 < \lambda_{\min}(\hat{\mathcal{L}}_2^{-1} \mathcal{K}_2)$$

then the matrix $\hat{\mathcal{L}}_3^{-1} \mathcal{K}_3$ is symmetric and positive definite in the scalar product $(u, v)_3 = (\mathcal{D}_3 u, v)_{\ell_2}$ with

$$\mathcal{D}_3 = \begin{pmatrix} \mathcal{D}_2 (\hat{\mathcal{L}}_2^{-1} \mathcal{K}_2 - \tau_2 I) & 0 \\ 0 & \hat{S}_3 \end{pmatrix}.$$

For the proof, see [62].
This theorem shows that we can apply the standard conjugate gradient method in the scalar product $(u, v)_3 = (\mathcal{D}_3 u, v)_{\ell_2}$ to the preconditioned system

$$\hat{\mathcal{L}}_3^{-1} \mathcal{K}_3 \, \underline{x} = \hat{\mathcal{L}}_3^{-1} \underline{b},$$

which is equivalent to (4.1), with $\underline{x} = (\underline{x}_1, \underline{x}_2, \underline{x}_3)^\top$ and $\underline{b} = (\underline{b}_1, \underline{b}_2, \underline{b}_3)^\top$. Then we obtain the following well–known error estimate for the n–th iterate $x^{(n)}$ of the conjugate gradient method in the corresponding energy norm:

$$\|x - x^{(n)}\|_{\mathcal{D}_3 \hat{\mathcal{L}}_3^{-1} \mathcal{K}_3} \leq \frac{2q^n}{1+q^{2n}} \|x - x^{(0)}\|_{\mathcal{D}_3 \hat{\mathcal{L}}_3^{-1} \mathcal{K}_3}$$

with $q = (\sqrt{\kappa(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3)} - 1)/(\sqrt{\kappa(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3)} + 1)$, where $\kappa(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3)$ denotes the relative condition number: $\kappa(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3) = \lambda_{\max}(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3)/\lambda_{\min}(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3)$.
The next theorem provides an estimate for this relative condition number depending on the quality of the preconditioners for the Schur complements:

Theorem 4.2 *If*

$$\underline{\sigma}_1\, \hat{S}_1 \leq S_1 \leq \overline{\sigma}_1\, \hat{S}_1, \quad \underline{\sigma}_2\, \hat{S}_2 \leq S_2 \leq \overline{\sigma}_2\, \hat{S}_2, \quad \underline{\sigma}_3\, \hat{S}_3 \leq S_3 \leq \overline{\sigma}_3\, \hat{S}_3$$

and the parameters τ_1 and τ_2 are chosen such that

$$\tau_1 < \underline{\lambda}_1, \quad \tau_2 < \underline{\lambda}_2$$

then the matrices $\hat{\mathcal{L}}_2^{-1}\mathcal{K}_2$ and $\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3$ are symmetric and positive definite with respect to the scalar products $(.,.)_2$ and $(.,.)_3$ and

$$\lambda_{\min}(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3) \geq \underline{\lambda}_3, \quad \lambda_{\max}(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3) \leq \overline{\lambda}_3 \quad \textit{and} \quad \kappa(\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3) \leq \frac{\overline{\lambda}_3}{\underline{\lambda}_3},$$

where $\underline{\lambda}_1 = \underline{\sigma}_1$,

$$\underline{\lambda}_{i+1} = \frac{1}{2\tau_i}\left(\overline{\sigma}_i(\underline{\sigma}_{i+1}+1) - \sqrt{\overline{\sigma}_i^2(\underline{\sigma}_{i+1}+1)^2 - 4\tau_i\overline{\sigma}_i\underline{\sigma}_{i+1}}\right) \quad \textit{for } i = 1, 2$$

and $\overline{\lambda}_3$ is the largest positive zero of the rational function

$$\overline{\theta}_3(\lambda) = -\frac{\lambda}{\overline{\sigma}_3} + \cfrac{1-\lambda}{-\cfrac{\tau_2\lambda}{\overline{\sigma}_2} + \cfrac{1-\tau_2\lambda}{-\cfrac{\tau_1\tau_2\lambda}{\overline{\sigma}_1} + 1}}.$$

For the proof see [62].

Remark 4.1 *1. It is necessary to keep the parameters τ_1 and τ_2 below the limiting values $\underline{\lambda}_1$ and $\underline{\lambda}_2$ in order to guarantee a well-defined scalar product $(u, v)_3$. On the other hand, the closer these parameters are to their limiting values, the better it is, i.e.: the smaller is the upper bound $\overline{\lambda}_3/\underline{\lambda}_3$ for the relative condition number.*

2. If the parameters τ_1 and τ_2 are chosen close to their limiting values $\underline{\lambda}_1$ and $\underline{\lambda}_2$ in the sense:

$$\frac{\underline{\lambda}_i}{\tau_i} = \mathcal{O}(1) \quad \textit{for } i = 1, 2,$$

and if the spectral bounds are reasonably scaled:

$$1 + \underline{\sigma}_i\overline{\sigma}_i \leq \mathcal{O}(\overline{\sigma}_i) \quad \textit{for } i = 1, 2, 3,$$

it can be shown, see [62], that

$$\frac{\overline{\lambda}_3}{\underline{\lambda}_3} = \mathcal{O}\left(\frac{\overline{\sigma}_1}{\underline{\sigma}_1}\frac{\overline{\sigma}_2}{\underline{\sigma}_2}\frac{\overline{\sigma}_3}{\underline{\sigma}_3}\right).$$

5 Data–Sparse BEM Preconditioners

As seen in the previous section we need to have appropriate efficient preconditioners for the local discrete single layer potentials $\widetilde{V}_{i,h}$, for the local discrete Schur complements $\widetilde{S}_{i,h}$ as well as for the assembled BETI Schur complement F, respectively.

Since the bilinear form of the local approximate discrete single layer potential $\widetilde{V}_{i,h}$ is spectrally equivalent to the Galerkin matrix $V_{i,h}$ and therefore to the $H^{-1/2}(\Gamma_i)$–norm, we may use any multilevel representation of an equivalent norm in $H^{-1/2}(\Gamma_i)$, [19, 47]. A suitable multilevel [6] operator is defined by

$$A_i^{-1/2} w_{i,h} \,=\, \sum_{j=0}^{J} h_{i,j}(Q_{i,j} - Q_{i,j-1}) w_{i,h}$$

for $w_{i,h} \in T_{i,h} = T_{i,J}$ where $T_{i,0} \subset T_{i,1} \subset \ldots \subset T_{i,J} = T_{i,h} \subset \ldots$ is a nested sequence of boundary element spaces of piecewise constant basis functions with related mesh sizes $h_{i,j} = \frac{1}{2} h_{i,j-1}$. Moreover, $Q_{i,j} : L_2(\Gamma_i) \to T_{i,j}$ is the L_2 Galerkin projection onto $T_{i,j}$.

Lemma 5.1 [47, Theorem 2] *The hold the spectral equivalence inequalities*

$$c_1^{V_i}\, J^{-2}\, \langle A_i^{-1/2} w_{i,h}, w_{i,h} \rangle_{\Gamma_i} \,\le\, (\widetilde{V}_{i,h} \underline{w}_i, \underline{w}_i) \,\le\, c_2^{V_i}\, \langle A_i^{-1/2} w_{i,h}, w_{i,h} \rangle_{\Gamma_i}$$

for all $w_{i,h} \in T_{i,J} = T_{i,h}$ with $J = \mathcal{O}(1 + |\log H/h|)$.

Note that in the case of piecewise constant basis functions the L_2 projections $Q_{i,j} w_i$ can be computed directly by inverting diagonal matrices.
The crucial issue in the multilevel preconditioning approach is the availability of the nested hierarchy of boundary element spaces $T_{i,j}$. Starting from a coarse mesh with an associated space $T_{i,0}$ one can define all boundary element spaces $T_{i,j}$ and therefore $T_{i,h} = T_{i,J}$ by a recursive refinement procedure. Instead of this more academic situation we now consider the case where already the boundary element space $T_{i,h} = T_{i,J}$ is given and no further refinement is applicable. In this case we have to construct an artificial mesh hierarchy to define an appropriate multilevel operator. For this again we will use ideas from fast boundary element methods, in particular we will introduce an appropriate clustering hierarchy of all boundary elements as it is used in the fast multipole method itself. Using a recursive bisection algorithm all boundary elements τ_k^i belonging to a subdomain boundary Γ_i can be clustered hierarchically. Since the boundary element spaces $T_{i,h}$ are spanned by piecewise constant basis functions, one can define corresponding piecewise constant coarse grid functions with respect to the cluster of underlying boundary elements. This results in an artificial hierarchy of piecewise constant trial spaces which can be used for the definition of the multilevel operator for preconditioning [53].

To construct preconditioning matrices C_{S_i} for the local discrete Schur complement matrices $\widetilde{S}_{i,h} = \widetilde{D}_{i,h} + \widetilde{K}_{i,h}^{\top} \widetilde{V}_{i,h}^{-1} \widetilde{K}_{i,h}$ we will apply the concept of boundary integral operators of the opposite order [56]. Based on the local trial space $U_{i,h} = S_h^1(\Gamma_i)$ of piecewise linear basis functions φ_m^i as used for the Galerkin discretization of the local hypersingular boundary integral operators D_i we define the Galerkin matrices

$$\bar{V}_{i,h}[n,m] \,=\, \langle \varphi_n^i, V \varphi_m^i \rangle_{\Gamma_i}, \quad \bar{M}_{i,h}[n,m] \,=\, \langle \varphi_n^i, \varphi_m^i \rangle_{\Gamma_i}$$

for $m, n = 1, \ldots, M_i$ and the application of the resulting preconditioning matrix is given by

$$C_{S_i}^{-1} \,=\, \bar{M}_{i,h}^{-1} \bar{V}_{i,h} \bar{M}_{i,h}^{-1} \quad \text{for } i = 1, \ldots, p. \tag{5.1}$$

When the application of the discrete single layer potential $\bar{V}_{i,h}$ is realized by using a fast multipole method as described in Section 2 this results in an approximated preconditioning matrix $\widetilde{C}_{S_i}$.

Lemma 5.2 *In the case of a non–floating subdomain* Ω_i *with* $\Gamma_i \cap \Gamma \neq \emptyset$ *there hold the spectral equivalence inequalities*

$$c_1^{S_i}\,[1+\log^2 H/h]^{-1}\,(\widetilde{C}_{S_i}\underline{v}_i,\underline{v}_i) \,\leq\, (\widetilde{S}_{i,h}\underline{v}_i,\underline{v}_i) \,\leq\, c_2^{S_i}(\widetilde{C}_{S_i}\underline{v}_i,\underline{v}_i) \tag{5.2}$$

for all $\underline{v}_i \in I\!R^{M_i}$ *while for a floating subdomain* Ω_i *with* $\Gamma_i \cap \Gamma = \emptyset$ *the spectral equivalence inequalities*

$$c_1^{S_i}\,(\widetilde{C}_{S_i}\underline{v}_i,\underline{v}_i) \,\leq\, (\widetilde{S}_{i,h}\underline{v}_i,\underline{v}_i) \,\leq\, c_2^{S_i}(\widetilde{C}_{S_i}\underline{v}_i,\underline{v}_i) \tag{5.3}$$

for all $\underline{v}_i \in I\!R^{M_i}$ *with* $\underline{e}_i^\top \underline{v}_i = 0$ *are valid where* $c_1^{S_i}$ *and* $c_2^{S_i}$ *are some positive constants independent of* h *and* H*.*

Proof. Since the approximated discrete single layer potential $\widetilde{V}_{i,h}$ is spectrally equivalent to the Galerkin matrix $V_{i,h}$ and therefore positive definite, and using the representation (2.22) of the approximate hypersingular boundary integral operator $\widetilde{D}_i$ we first conclude

$$\begin{aligned}
(\widetilde{S}_{i,h}\underline{v}_i,\underline{v}_i) \;&\geq\; (\widetilde{D}_{i,h}\underline{v}_i,\underline{v}_i) + (\widetilde{V}_{i,h}^{-1}\widetilde{K}_{i,h}\underline{v}_i,\widetilde{K}_{i,h}\underline{v}_i) \,\geq\, (\widetilde{D}_{i,h}\underline{v}_i,\underline{v}_i)\\
&=\; \sum_{k=1}^{3}\langle \widetilde{V}_{i,h}\underline{\mathbf{curl}}_{|\Gamma_i} v_{i,h|k}, \underline{\mathbf{curl}}_{|\Gamma_i} v_{i,h|k}\rangle_{\Gamma_i}\\
&\geq\; c\sum_{k=1}^{3}\langle V_i\underline{\mathbf{curl}}_{|\Gamma_i} v_{i,h|k}, \underline{\mathbf{curl}}_{|\Gamma_i} v_{i,h|k}\rangle_{\Gamma_i} \;=\; c\,(D_{i,h}\underline{v}_i,\underline{v}_i)
\end{aligned}$$

for all $\underline{v}_i \in I\!R^{M_i}$. The approximate discrete Steklov–Poincaré operator $\widetilde{S}_{i,h}$ is bounded above [9, 54], i.e.

$$(\widetilde{S}_{i,h}\underline{v}_i,\underline{v}_i) \,\leq\, (S_{i,h}\underline{v}_i,\underline{v}_i) \,=\, \langle S_i v_{i,h}, v_{i,h}\rangle_{\Gamma_i} \,\leq\, c\,\langle D_i v_{i,h}, v_{i,h}\rangle_{\Gamma_i} \,=\, c\,(D_{i,h}\underline{v}_i,\underline{v}_i)$$

holds for all $\underline{v}_i \in I\!R^{M_i}$. Hence, to find a preconditioner for $\widetilde{S}_{i,h}$ it is sufficient to find a preconditioner for the discrete hypersingular integral operator $D_{i,h}$. In the case of a floating subdomain Ω_i with $\Gamma_i \cap \Gamma = \emptyset$ the preconditioning matrix $C_{S_i} = \bar{M}_{i,h}\bar{V}_{i,h}M_{i,h}$ is spectrally equivalent to the discrete hypersingular integral operator $D_{i,h}$ [56], and since the multipole approximation of $\bar{V}_{i,h}$ is spectrally equivalent to $\bar{V}_{i,h}$ the same is true for the approximate preconditioner $\widetilde{C}_{S_i}$, see also [45]. Hence we obtain the spectral equivalence inequalities (5.3).
When the subdomain Ω_i is a non–floating subdomain with $\Gamma_i \cap \Gamma \neq \emptyset$ we have to use the trial space $U_{i,h} \subset \widetilde{H}^{1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i))$ of piecewise linear basis functions vanishing on the Dirichlet boundary Γ. Due to the mapping properties of the boundary integral operators

$$\begin{aligned}
D_i \;&:\; \widetilde{H}^{1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i)) \to H^{-1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i)),\\
V_i \;&:\; \widetilde{H}^{-1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i)) \to H^{1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i))
\end{aligned}$$

we have to consider the discrete embedding

$$\|v_h\|^2_{H^{1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i))} \,\leq\, c\,[1+\log^2 H/h]\,\|v_h\|^2_{\widetilde{H}^{1/2}(\Gamma_i\backslash(\Gamma\cap\Gamma_i))}$$

for $v_h \in U_{i,h}$ yielding the spectral equivalence inequalities (5.2), see [43] together with an appropriate scaling argument. ■

We finally have to describe a suitable preconditioning matrix for the BETI Schur complement

$$F \;=\; \sum_{i=1}^{q}\frac{1}{\alpha_i}B_i[\widetilde{S}_{i,h}+\beta_i\underline{e}_i\underline{e}_i^\top]^{-1}B_i^\top + \sum_{i=q+1}^{p} B_i\widetilde{S}_{i,h}^{-1}B_i^\top\,.$$

Following [38] we can define the scaled sparse hypersingular BETI preconditioner

$$C_F^{-1} = (BC_\alpha^{-1}B^T)^{-1}BC_\alpha^{-1}\widetilde{D}_hC_\alpha^{-1}B^T(BC_\alpha^{-1}B^\top)^{-1}. \tag{5.4}$$

Lemma 5.3 [38, Theorem 3.2] *For the scaled sparse hypersingular BETI preconditioner* (5.4)*, the condition estimate*

$$\kappa(PC_F^{-1}P^TP^TFP) \leq c\left(1+\log\frac{H}{h}\right)^2 \tag{5.5}$$

holds, where the positive constant c is independent of h, H, p and the α_i's (coefficient jumps).

Combining Theorem 4.2 with the results of Lemmata 5.1–5.3 and taking into account the complexity estimate for the fast multipole method we can finally arrive at our main theorem.

Theorem 5.1 *If the two–fold saddle point problem (3.13) is solved by the 2SPCG algorithm where the preconditioner is built from the block preconditioners C_V, C_S and C_F, respectively, then not more than $I(\varepsilon) = \mathcal{O}((1+\log(H/h))^3 \log\varepsilon^{-1})$ iterations and $ops(\varepsilon) = \mathcal{O}((H/h)^2(1+\log(H/h))^5\log\varepsilon^{-1})$ arithmetical operations in a parallel regime are required in order to reduce the initial error by the factor $\varepsilon\in(0,1)$ in a parallel regime. The number of iterations $I(\varepsilon)$ is robust with respect to the jumps in the coefficients. Moreover, not more than $\mathcal{O}((H/h)^2(1+\log(H/h))^2)$ storage units are needed per processor.*

Note that the complexity estimates given in Theorem 5.1 take not into account the costs for the projection (3.11) that involves the global information exchange.

Remark 5.1 *If we use optimal preconditioners $\widetilde{C}_{V_i}$ for the local single layer potentials $\widetilde{V}_{i,h}$ and $\widetilde{C}_{S_i}$ for the local boundary element Schur complements $\widetilde{S}_{i,h}$, then the number of iteration $I(\varepsilon)$ of our 2SPCG solver would behave like $\mathcal{O}((1+\log(H/h))\log\varepsilon^{-1})$, whereas the arithmetical complexity would decrease from $\mathcal{O}((H/h)^2(1+\log(H/h))^5\log\varepsilon^{-1})$ to $\mathcal{O}((H/h)^2(1+\log(H/h))^3\log\varepsilon^{-1})$. Such preconditioners are available, see, e.g., [37, 49, 56, 57]. If we convert the non–floating subdomains having a Dirichlet boundary part to floating subdomains by including the Dirichlet boundary condition into the constraints, then the data–sparse opposite order preconditioners $\widetilde{\mathcal{S}}_{i,h}$ given above is optimal.*

Applying the 2SPCG introduced in Section 4 to our two–fold saddle point problem (3.13) with the preconditioners presented above and taking into account Remark 3.1, we obtain a subspace iteration version of the 2SPCG that can be rewritten in following algorithmic form:

Algorithm 5.1 Inexact BETI two–fold saddle point problem CG

$\underline{\lambda}_e = QG^\top(G^\top QG)^{-1}\underline{e}$ {forcing the constraint $G^\top\underline{\lambda}_e = \underline{e}$}
$\underline{x}_1^0 = \underline{0}, \underline{x}_2^0 = \underline{0}, \underline{x}_3^0 = \underline{\lambda}_e$ {choose the initial guess}
$\underline{b}_1 = \underline{g}, \underline{b}_2 = \underline{d} = \underline{f} - B^\top\underline{\lambda}_e, \underline{b}_3 = \underline{0}$ {compute right hand side}
$\underline{d} = \mathcal{K}_3\underline{x}^0 - \underline{b}$ {compute the defect}
$\underline{r}^0 = \mathcal{D}_3\hat{\mathcal{L}}_3^{-1}\underline{d}$ {apply the transformation}
$\underline{p}^0 = \underline{w}^0 = \mathcal{D}_3^{-1}\underline{r}^0$ {preconditioning}
$\rho_0 = (\underline{w}^0, \underline{r}^0)$
for $n=0$ **step** 1 **until** $\rho_n \leq \varepsilon\rho_0$ **do**
  $\underline{v}^n = \mathcal{D}_3\hat{\mathcal{L}}_3^{-1}\mathcal{K}_3\underline{p}^n$ {matrix vector multiplication and transformation}
  $\sigma = (\underline{v}^n, \underline{p}^n)$
  $\alpha = \rho_{n+1}/\sigma$
  $\underline{x}^{n+1} = \underline{x}^n - \alpha\underline{p}^n$ {update the iterate}
  $\underline{r}^{n+1} = \underline{r}^n - \alpha\underline{v}^n$ {update the defect}
  $\underline{w}^{n+1} = \mathcal{D}_3^{-1}\underline{r}^{n+1}$ {preconditioning}
  $\rho_{n+1} = (\underline{w}^{n+1}, \underline{r}^{n+1})$
  $\beta = \rho_{n+1}/\rho_n$
  $\underline{p}^{n+1} = \underline{w}^{n+1} + \beta\underline{p}^n$ {update of the search direction}
end for

Note that for an efficient implementation, the application of $\mathcal{D}_3\hat{\mathcal{L}}_3^{-1}$ and $\mathcal{D}_3^{-1}$ have to be concatenated, see [55, 62]. Here, we have used the notations of Section 4 for the matrix in (3.13) and a vector $\underline{x} = (\underline{x}_1, \underline{x}_2, \underline{x}_3)^T$, where $\underline{x}_i$ corresponds to the i-th block of the system (3.13).
Algorithm 5.1 was used in our numerical experiments the results of which are given in the next section.

6 Numerical Results

For the numerical examples we consider the unit cube which is subdivided into eight similar subdomains as shown in Figure 1. The given Dirichlet data $g(x)$ is chosen as the trace of a regular solution of the boundary value problem (2.1). This allows to check the convergence of the computed numerical solution to the exact one.

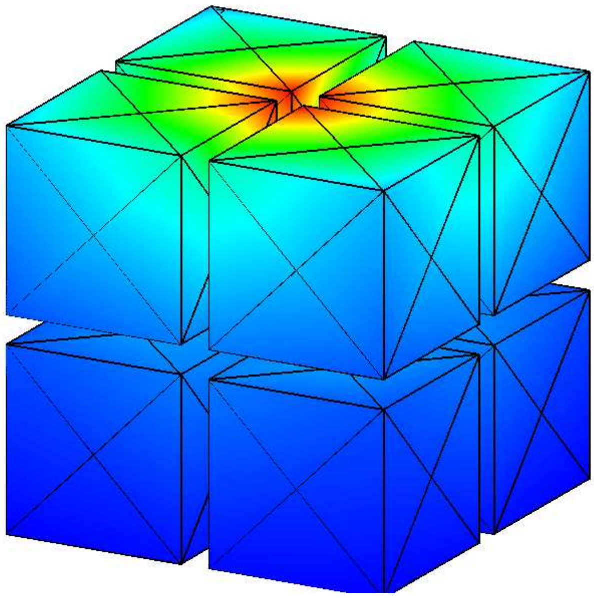

Figure 1: Domain decomposition with 8 subdomains

In Table 1, the geometric informations of the domain and the subdomains are listed for the refinement levels L. Starting from the coarsest grid with 192 triangles for the whole domain Ω, the refined meshes are recursively constructed by subdividing each triangle into four smaller similar triangles. N and M are the total numbers of triangles, respectively nodes of the whole domain. M_c is the total number of coupling nodes. The numbers of local triangles and nodes of a single subdomain Ω_i are given by N_i and M_i, respectively. If the boundary mesh of one subdomain Ω_i with 98 304 triangles was uniformly extended to the interior of the subdomain, the corresponding FEM grid would consist of 4448731 tetrahedra. For the whole domain almost 36 millions tetrahedra would be used.
In all following tables, a unified notation is used. L again denotes the refinement level and Table 1 gives the corresponding information of the grids. t_1 and t_2 are the measured times in seconds for setting up the respective system of linear equations and for their solution. *it* denotes the number of iterations needed to solve the respective system of linear equations with a relative accuracy of 10^{-8}. In the column *error* the error $\|u - u_h\|_{L_2(\Gamma_i)}$ is itemized for the different levels.
The effects of the preconditioners for inverting the single layer potential in the representation of the Steklov–Poincare operator and for the Schur complement system (2.25) by the preconditioner (2.26) are shown in Table 2. Setting up the preconditioners takes some time as can be seen in the

L	N	M	M_c	N_i	M_i
0	192	63	13	24	14
1	768	261	67	96	50
2	3072	1089	319	384	194
3	12288	4473	1399	1536	770
4	49152	18153	5863	6144	3074
5	196608	73161	24007	24576	12290
6	786432	293769	97159	98304	49154

Table 1: Geometric information

	unpreconditioned				preconditioned			
L	t_1	t_2	it	error	t_1	t_2	it	error
0	0	0	4(6)	3.2954e-2	0	0	4(11)	3.2954e-2
1	1	0	12(21)	5.1662e-3	1	0	14(22)	5.1662e-3
2	2	3	17(33)	1.1907e-3	4	3	20(28)	1.1907e-3
3	9	30	25(46)	2.8996e-4	13	19	22(31)	2.8996e-4
4	35	380	36(65)	6.9990e-5	48	150	24(35)	6.9989e-5
5	159	4912	52(87)	1.7508e-5	178	1194	26(38)	1.7506e-5
6	807	40294	74(122)	4.3532e-6	844	6099	29(41)	4.3488e-6

Table 2: Schur complement system of (2.16)

differences of t_1 for the unpreconditioned and preconditioned case. But the preconditioning pays off as the numbers *it* of iterations are reduced significantly. In brackets the average numbers of iterations are given for the inversion of the local single layer potentials. Accordingly, the times t_2 for solving the system of linear equations are reduced, while the errors are about the same. These calculations have been executed on a standard PC pool with 3.06 MHz Intel processors and 1 GB of RAM.
Next, the Schur complement system (2.25) will be compared to three different BETI formulations. As the times t_1 to set up the system of linear equations only differ slightly from each other due to measuring inaccuracies, only the times t_2 for solving and the numbers *it* of iterations will be compared in Table 3. Only the numbers for the preconditioned systems are treated. The numbers of iterations differ slightly from those of Table 2, as a slightly different preconditioner C_V^{-1} was used for the inversion of the local single layer potentials $\widetilde{V}_{i,h}$. The computational times are a little bit higher than in the example of Table 2 as the computations have been executed on a different cluster. All further computations have been executed on the mozart cluster of the Chair of Simulation of Large Systems and the Chair of Numerics for Supercomputers at the University of Stuttgart. The cluster consists of 64 nodes with 2 CPU Intel Xeon 3,066 and 4 GB RAM each. In addition, Table 4 itemizes the corresponding errors.

	(2.25)		(3.16)		(3.15)		(3.3)	
L	t_2	it	t_2	it	t_2	it	t_2	it
0	0	4(12)	2	2(3(12))	2	9(12)	1	40
1	2	14(22)	6	6(10(22))	4	27(22)	3	67
2	4	20(28)	33	11(16(28))	11	32(28)	12	84
3	24	22(32)	263	12(18(32))	77	36(32)	70	99
4	164	24(36)	2221	14(21(36))	572	41(35)	450	111
5	1427	26(40)	21429	15(23(39))	4947	44(38)	3739	128
6	7691	29(43)	132892	17(26(43))	26011	48(42)	18875	148

Table 3: comparison of standard DD and BETI for 8 cubes

The first BETI formulation is the Schur complement system (3.16). This system involves two recursive inversions in each iteration step. The iteration numbers given in brackets are average numbers needed to invert the local Steklov-Poincare operators and the local single layer potential, respectively, while the other number is the number of iterations for solving the BETI Schur complement system (3.16). For example, on the sixth refinement level about $17 \cdot 26$ inversions of the local single layer potentials are executed and the average number of iterations needed for such an inversions is 43.
The second BETI formulation is the saddle point problem (3.15). This system only needs the inversions of the local single layer potentials in each iteration step of the system, while the local potentials $\underline{u}_i$ are determined simultaneously in the global system. Due to the larger number of unknowns in the global system, the number of iterations raises but not too strongly. Therefore the total number of applications of the Galerkin matrices of the local boundary integral operators is less than for the BETI Schur complement system. Correspondingly, the improvement on the times t_2 for solving the system is significant.
A further improvement on the times t_2 can be made by using the third BETI formulation, the twofold saddle point problem given by (3.3). In this formulation no inner inversion is needed, but the number of unknowns in the system is raised by almost 800 000 for the local fluxes $\underline{t}_i$ on the sixth level. Therefore the number of iterations again raises. Correspondingly, the number of applications of the Galerkin matrices of the local single layer potentials is again decreased. The benefit for solving the system is lower than in the case before, as the number of application of the Galerkin matrices of the other boundary integral operator raises. Unfortunately, theses applications are more expensive. The formulation (3.3) using the twofold saddle point problem performs best of all BETI formulations. But it still performs worse than the Schur complement system (2.25).
The systems (3.15) and (3.3) have been solved as the transformed systems described in Section 4 by using Algorithm 5.1. The used preconditioners have been scaled optimally according to the estimates of the eigenvalues of the transformed systems given by Theorem 4.2.
In Table 4 the errors $\|u - u_h\|_{L_2(\Gamma_S)}$ are compared for the miscellaneous approaches. As can be easily seen, there are not small differences in the accuracy for the different formulations. The two–fold saddle point formulation (3.3) seems to give the best approximation.

L	(2.25)	(3.16)	(3.15)	(3.3)
0	3.2954e-2	3.2954e-2	3.2954e-2	3.2954e-2
1	5.1662e-3	5.1662e-3	5.1662e-3	5.1662e-3
2	1.1907e-3	1.1907e-3	1.1909e-3	1.1907e-3
3	2.8996e-4	2.8996e-4	2.9012e-4	2.8995e-4
4	6.9990e-5	6.9993e-5	7.0097e-5	6.9954e-5
5	1.7506e-5	1.7510e-5	1.7580e-5	1.7466e-5
6	4.3487e-6	4.3514e-6	4.3894e-6	4.2888e-6

Table 4: comparison of the errors for standard DD and BETI

The BETI methods perform much better when considering jumping coefficients instead of constant coefficients as in the previous example. We have chosen a different regular solution which fits also to jumping coefficients. In Table 5 the time t_2 for solving the linear system and the corresponding numbers *it* of iterations are itemized, while in Table 6 the corresponding numbers are itemized for the coefficients $\alpha_1 = 1$ and $\alpha_2 = 10^5$ distributed on the subdomains like on a chequerboard. Again, the three different BETI formulations are compared to the Schur complement system (2.25).

	(2.25)		(3.16)		(3.15)		(3.3)	
L	t_2	it	t_2	it	t_2	it	t_2	it
0	0	4(12)	2	2(3(12))	2	9(12)	1	40
1	2	14(22)	6	6(10(22))	4	27(22)	3	67
2	4	20(28)	33	11(16(28))	11	32(28)	12	84
3	24	22(32)	263	12(18(32))	77	36(32)	70	99
4	164	24(36)	2221	14(21(36))	572	41(35)	450	111
5	1427	26(40)	21429	15(23(39))	4947	44(38)	3739	128
6	7691	29(43)	132892	17(26(43))	26011	48(42)	18875	148

Table 5: comparison of standard DD and BETI

	(2.25)		(3.16)		(3.15)		(3.3)	
L	t_2	it	t_2	it	t_2	it	t_2	it
0	2	8(12)	2	2(2(4))	1	5(4)	1	30
1	2	16(19)	4	6(6(18))	4	17(18)	2	43
2	5	23(27)	29	10(14(28))	10	25(26)	8	58
3	29	28(32)	245	12(16(32))	70	32(31)	51	71
4	226	33(36)	1929	13(19(36))	497	36(34)	328	80
5	2439	44(39)	17762	13(21(39))	4442	40(37)	2593	89
6	15045	55(44)	106341	14(23(43))	24748	45(41)	13667	106

Table 6: comparison of standard DD and BETI for jumping coefficients

Comparing the iteration numbers in Table 5 and 6, the scaled version of the BETI preconditioner performs even better and this example confirms the independence of the preconditioned system of the respective coefficients. On the other hand the iteration numbers for solving the Schur complement system of (2.16) increase in the case of jumping coefficients, even though the preconditioner still works.

Comparing the times t_2 for solving the respective systems, the numbers are a lot better for the BETI method. The twofold saddle point formulation (3.3) of the BETI method outperforms the

Schur complement system (2.25) on the sixth refinement level and one would expect an even larger difference of the seventh level.

7 Conclusions and Final Remarks

In this paper we presented inexact data–sparse Boundary Element Tearing and Interconnecting domain decomposition methods for solving boundary value problems for the potential equation with piecewise constant coefficients. In contrast to the classical approach we avoided the elimination of local unknowns, i.e. of the subdomain vectors $\underline{t}_i$ and $\underline{u}_i$. This finally led us to a two–fold saddle point problem. We presented preconditioned CG–like iterative methods for solving such kind of two–fold saddle point problems and gave an rigorous analysis of the preconditioning problem resulting in precise CG–like convergence rate estimates. Furthermore, we present data–sparse preconditioners which yield an almost optimal iterative solver for the inexact data–sparse BETI equations. Moreover, the convergence rate is not affected by large jumps in the coefficients of the PDE. This is confirmed by all our numerical experiments. The data–sparse preconditioners proposed in this paper do not need special constructions since they are available in the BETI discretization anyway. So, they are completely algebraic. This is very important for the practical acceptance of our method. It is clear that inexact data–sparse BETI allows us to solve really large–scale problems with several million boundary unknowns on distributed memory parallel computers very efficiently.

The treatment of the outer Dirichlet problem and other boundary conditions is straightforward. The generalization of our method to linear elasticity problems with piecewise constant materials is also possible. Using the symmetric coupling technique proposed by Costabel [10] and combining the results of this paper with the results obtained by Klawonn and Widlund [33], we can easily construct and analyze inexact FETI–BETI solvers for large scale coupled finite and boundary element equations. Exact FETI–BETI solvers were already proposed and analyzed in [39].

References

[1] M. Bebendorf: Approximation of boundary element matrices. Numer. Math. 86 (2000) 565–589.

[2] M. Bebendorf: Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang–Matrizen. Doctoral Thesis, Universität des Saarlandes, 2000.

[3] M. Bebendorf, S. Rjasanow: Adaptive low–rank approximation of collocation matrices. Computing 70 (2003) 1–24.

[4] C. Börgers: The Neumann–Dirichlet domain decomposition method with inexact solvers on the subdomains. Numer. Math. 55 (1989) 123–136.

[5] J. H. Bramble, J. E. Pasciak: A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. Math. Comp. 50 (1988) 1–17.

[6] J. H. Bramble, J. E. Pasciak, J. Xu: Parallel multilevel preconditioners. Math. Comp. 55 (1990) 1–22.

[7] S. C. Brenner: An additive Schwarz preconditioner for the FETI method. Numer. Math. 94 (2003) 1–31.

[8] J. Carrier, L. Greengard, V. Rokhlin: A fast adaptive multipole algorithm for particle simulations. SIAM J. Sci. Stat. Comput. 9 (1988) 669–686.

[9] C. Carstensen, M. Kuhn, U. Langer: Fast parallel solvers for symmetric boundary element domain decomposition equations. Numer. Math. 79 (1998) 321–347.

[10] M. Costabel: Symmetric methods for the coupling of finite elements and boundary elements. In: Boundary Elements IX (C. A. Brebbia, G. Kuhn, W. L. Wendland eds.), Springer, Berlin, pp. 411–420, 1987.

[11] M. Costabel: Boundary integral operators on Lipschitz domains: Elementary results. SIAM J. Math. Anal. 19 (1988) 613–626.

[12] W. Dahmen, S. Prössdorf, R. Schneider: Wavelet approximation methods for pseudodifferential equations II: Matrix compression and fast solution. Adv. Comput. Math. 1 (1993) 259–335.

[13] C. Farhat, M. Lesoinne, A. P. Macedo: A two–level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. Numer. Math. 85 (2000) 283–308.

[14] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, D. Rixen: FETI–DP: A dual–primal unified FETI method I: A faster alternative to the two–level FETI method. Int. J. Numer. Meth. Engrg. 50 (2001) 1523–1544.

[15] C. Farhat, M. Lesoinne, K. Pierson: A scalable dual–primal domain decomposition method. Numer. Linear Algebra Appl. 7 (2000) 687–714.

[16] C. Farhat, F.–X. Roux: A method of finite element tearing and interconnecting and its parallel solution algorithm. Int. J. Numer. Meth. Engrg. 32 (1991) 1205–1227.

[17] C. Farhat, F.–X. Roux: Implicit parallel processing in structural mechanics. Comput. Mech. Adv. 2 (1994) 1–124.

[18] Y. Fragakis, M. Papadrakakis: A unified framework for formulating domain decomposition methods in structural mechanics. Technical Report, National Technical University, Athen, 2002.

[19] S. A. Funken, E. P. Stephan: The BPX preconditioner for the single layer potential operator. Appl. Anal. 67 (1997) 327–340.

[20] G. N. Gatica, N. Heuer: Conjugate gradient method for dual–dual mixed formulations. Math. Comp. 71 (2002) 1455–1472.

[21] L. Greengard: The Rapid Evaluation of Potential Fields in Particle Simulation. The MIT Press, Cambridge, MA, 1987.

[22] L. Greengard, V. Rokhlin: A fast algorithm for particle simulations. J. Comput. Phys. 73 (1987) 325–348.

[23] G. Haase, B. Heise, M. Kuhn, U. Langer: Adaptive domain decomposition methods for finite and boundary element equations. In: Boundary Element Topics (W. L. Wendland ed.), Springer, Berlin, pp. 121–147, 1998.

[24] G. Haase, U. Langer, A. Meyer: The approximate Dirichlet domain decomposition method. Computing 47 (1991) 137–167.

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

[26] G. Haase, S. Nepomnyaschikh: Explicit extension operators on hierarchical grids. East–West J. Numer. Math. 5 (1997) 231–248.

[27] W. Hackbusch: A sparse matrix arithmetic based on $\mathcal{H}$–matrices. Computing 62 (1999) 89–108.

[28] W. Hackbusch, Z. P. Nowak: On the fast matrix multiplication in the boundary element method by panel clustering. Numer. Math. 54 (1989) 463–491.

[29] E. W. Hobson: The Theory of Spherical and Ellipsoidal Harmonics. Chelsea, New York, 1955.

[30] G. C. Hsiao, O. Steinbach, W. L. Wendland: Domain decomposition methods via boundary integral equations. J. Comput. Appl. Math. 125 (2000) 521–537.

[31] G. C. Hsiao, W. L. Wendland: Domain decomposition in boundary element methods. In: Proceedings of the Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Periaux, O. B. Widlund eds.), SIAM, Philadelphia, pp. 41–49, 1991.

[32] M. Kaltenbacher: Numerical Simulation of Mechatronic Sensors and Actuators. Springer, Heidelberg, 2004.

[33] A. Klawonn, O. B. Widlund: A domain decomposition method with Lagrange multipliers and inexact solvers for linear elasticity. SIAM J. Sci. Comput. 22 (2000) 1199–1219.

[34] A. Klawonn, O. B. Widlund: FETI and Neumann–Neumann iterative substructuring methods: Connections and new results. Comm. Pure Appl. Math. 54 (2001) 57–90.

[35] A. Klawonn, O. B. Widlund, M. Dryja: Dual–primal FETI methods for three–dimensional elliptic problems with heterogeneous coefficients. SIAM J. Numer. Anal. 40 (2002) 159–179.

[36] U. Langer: Parallel iterative solution of symmetric coupled FE/BE–equations via domain decomposition. Contemp. Math. 157 (1994) 335–344.

[37] U. Langer, D. Pusch, S. Reitzinger: Efficient preconditioners for boundary element matrices based on grey–box algebraic multigrid methods. Int. J. Numer. Methods Engrg. 58 (2003) 1937–1953.

[38] U. Langer, O. Steinbach: Boundary element tearing and interconnecting methods. Computing 71 (2003) 205–228.

[39] U. Langer, O. Steinbach: Coupled boundary and finite element tearing and interconnecting methods. In: Proceedings of the 15th International Conference on Domain Decomposition (R. Kornhuber, R. Hoppe, J. Periaux, O. Pironneau, O. Widlund, J. Xu eds.), Lecture Notes in Computational Sciences and Engineering, vol. 40, Springer, Heidelberg, pp. 83–97, 2004.

[40] J. Mandel, R. Tezaur: Convergence of a substructuring method with Lagrange multipliers. Numer. Math. 73 (1996) 473–487.

[41] J. Mandel, R. Tezaur: On the convergence of a dual–primal substructuring method. Numer. Math. 88 (2001) 543–558.

[42] W. McLean: Strongly Elliptic Systems and Boundary Integral Equations. Cambridge University Press, 2000.

[43] W. McLean, O. Steinbach: Boundary element preconditioners for a hypersingular integral equation on an interval. Adv. Comput. Math. 11 (1999) 271–286.

[44] J. Nédélec: Integral equations with non integrable kernels. Int. Eq. Operat. Th. 5 (1982) 562–572.

[45] G. Of, O. Steinbach, A fast multipole boundary element method for a modified hypersingular boundary integral equation. In: Proceedings of the International Conference on Multifield Problems (M. Efendiev, W. L. Wendland eds.). Springer Lecture Notes in Applied Mechanics, vol. 12, Springer, Berlin, 2003, 163–169.

[46] G. Of, O. Steinbach, W. L. Wendland: The fast multipole method for the symmetric boundary integral formulation. Bericht 2004/08, SFB 404, Universität Stuttgart, 2004.

[47] P. Oswald: Multilevel norms in $H^{-1/2}(\Gamma)$. Computing 61 (1998) 235–255.

[48] J. M. Perez–Jorda, W. Yang: A concise redefinition of the solid spherical harmonics and its use in the fast multipole method. J. Chem. Phys. 104 (1996) 8003–8006.

[49] T. v. Petersdorff, E. P. Stephan: Multigrid solvers and preconditioners for first kind integral equations. Num. Methods Part. Diff. Eq. 8 (1992) 443–450.

[50] K. H. Pierson, P. Raghaven, G. M. Reese: Experiences with FETI–DP in a production level finite element application. In: Proceedings of the 14th International Conference on Domain Decomposition Methods (I. Herrera, D. E. Keyes, O. Widlund, R. Yates eds.), DDM.org, pp. 233–240, 2003.

[51] V. Rokhlin: Rapid solution of integral equations of classical potential theory . J. Comput. Phys. 60 (1985) 187–207.

[52] D. Stefanica: A numerical study of FETI algorithms for mortar finite element methods. SIAM J. Sci. Comput. 23 (2001) 1135–1160.

[53] O. Steinbach: Artificial multilevel boundary element preconditioners. Proc. Appl. Math. Mech. 3 (2003) 539–542.

[54] O. Steinbach: Stability estimates for hybrid coupled domain decomposition methods. Lecture Notes on Mathematics, vol. 1809, Springer, Heidelberg, 2003.

[55] O. Steinbach: Numerische Näherungsverfahren für elliptische Randwertprobleme. Finite Elemente und Randelemente. B. G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2003.

[56] O. Steinbach, W. L. Wendland: The construction of some efficient preconditioners in the boundary element method. Adv. Comput. Math. 9 (1998) 191–216.

[57] E. P. Stephan: Multilevel methods for the h–, p–, and hp–versions of the boundary element method. J. Comput. Appl. Math. 125 (2000) 503–519.

[58] A. Toselli, O. Widlund: Domain Decomposition Methods – Algorithms and Theory. Springer Series in Computational Mathematics, vol. 34, Springer, New York, 2004.

[59] C. A. White, M. Head–Gordon: Derivation and efficient implementation of the fast multipole method. J. Chem. Phys. 101 (1994) 6593–6605.

[60] K. Yosida: Applications of fast multipole method to boundary integral equation method. Dept. of Global Environment Eng., Kyoto Univ., Japan, 2001.

[61] W. Zulehner: Analysis of iterative methods for saddle point problems: A unified approach. Math. Comp. 71 (2002) 479–505.

[62] W. Zulehner: Uzawa–type methods for block–structured indefinite linear systems. SFB–Report 2005-05, SFB F013, Johannes Kepler University Linz, Austria, 2005.

Erschienene Preprints ab Nummer 2005/1

2005/1	O. Steinbach	Numerische Mathematik 1. Vorlesungsskript.
2005/2	O. Steinbach	Technische Numerik. Vorlesungsskript.
2005/3	U. Langer G. Of O. Steinbach W. Zulehner	Inexact Fast Multipole Boundary Element Tearing and Interconnecting Methods
2005/4	U. Langer G. Of O. Steinbach W. Zulehner	Inexact Data–Sparse Boundary Element Tearing and Interconnecting Methods