

Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Accelerating Galerkin BEM for Linear Elasticity
using Adaptive Cross Approximation

by

Mario Bebendorf, and Richard Grzibovski

Preprint no.: 46

2006



Accelerating Galerkin BEM for Linear Elasticity using Adaptive Cross Approximation

M. Bebendorf* and R. Grzibovski†

May 2, 2006

The adaptive cross approximation (ACA) algorithm [2, 3] provides a means to compute data-sparse approximants of discrete integral formulations of elliptic boundary value problems with almost linear complexity. ACA uses only few of the original entries for the approximation of the whole matrix and is therefore well-suited to speed up existing computer codes. In this article we extend the convergence proof of ACA to Galerkin discretizations. Additionally, we prove that ACA can be applied to integral formulations of systems of second-order elliptic operators without adaptation to the respective problem. The results of applying ACA to boundary integral formulations of linear elasticity are reported. Furthermore, we comment on recent implementation issues of ACA for nonsmooth boundaries.

AMS Subject Classification: 65D05, 65D15, 65F05, 65F30

Keywords: integral equations, hierarchical matrices, fast solvers

1 Introduction

Due to improved capabilities of recent computer systems, larger and larger linear systems of equations are to be solved. Since classical solution techniques such as Gaussian elimination have superlinear complexity, the efficiency of such methods gets worse the faster computers develop. Therefore, in the past few years fast summation methods have become widely popular in the field of engineering and applied mathematics.

In this article we are interested in the efficient solution of systems of integral equations

$$\lambda u(y) + (\mathcal{K}u)(y) = f(y), \quad y \in \Gamma, \quad (1.1)$$

with integral operators $(\mathcal{K}u)(y) := \int_{\Gamma} \kappa(x, y)u(x) ds_x$ and given right-hand sides f . The solution $u : \Gamma \rightarrow \mathbb{R}^{\nu}$ is searched for on a $(d-1)$ -dimensional manifold $\Gamma \subset \mathbb{R}^d$. This kind of integral equation arises for instance from the boundary element method (BEM); cf. [24, 20]. The results of this article can however be easily extended to volume integral equations in \mathbb{R}^d .

In order to solve equation (1.1) numerically, the domain of integration Γ is approximated by a triangulation Γ_h . Besides the *Galerkin method*, the *collocation method*, and the *Nyström method* are

*Fakultät für Mathematik und Informatik, University of Leipzig, Augustusplatz 10/11, D-04109 Leipzig

†Fachbereich Mathematik, Saarland University, Im Stadtwald, D-66123 Saarbrücken

commonly used. In the Galerkin and the collocation method the solution u is approximated from a finite-dimensional ansatz space V_h ; i.e., the approximant $u_h \in V_h$ to u is sought of the form

$$u_h(x) = \sum_{i=1}^N u_i \varphi_i(x),$$

where φ_i , $i \in I := \{1, \dots, N\}$, is a basis of V_h . Especially for complicated domains in three dimensions, the number of degrees of freedom N has to be chosen sufficiently large to guarantee a reasonable accuracy of u_h . All three methods reduce (1.1) to a linear system of the form

$$(\lambda M + K)x = b, \quad K, M \in \mathbb{R}^{I \times I}, \quad b \in \mathbb{R}^I,$$

where M is a sparse matrix and produces no numerical difficulties. In this article M will hence be neglected. However, K is dense and therefore requires $\mathcal{O}(N^2)$ units of storage. Additionally, the computation of the right-hand side b might involve the treatment of rectangular matrices. Therefore, we are concerned with matrices $A \in \mathbb{R}^{I \times J}$ with $J := \{1, \dots, N'\}$, which in the case of the Petrov-Galerkin method are

$$a_{ij} = \int_{\Gamma} \int_{\Gamma} \kappa(x, y) \varphi_j(x) \psi_i(y) \, ds_x \, ds_y$$

with trial functions ψ_i , in the case of the collocation method

$$a_{ij} = \int_{\Gamma} \kappa(x, y_i) \varphi_j(x) \, ds_x$$

with collocation points y_i , and in the case of the Nyström method

$$a_{ij} = \omega_j \kappa(y_j, y_i) \quad \text{for } i \neq j \quad \text{and} \quad a_{ii} = c_i$$

with N numbers c_i and weights ω_j . Integral equations are just one example of problems which lead to coefficient matrices of one of the above types. The evaluation of Coulomb and gravitational potentials of particle systems are further examples.

Throughout this article we will use the notation $\mathbb{R}^{t \times s}$ for the subspace of matrices in the rows $t \subset I$ and columns $s \subset J$. Furthermore, A_{ts} or A_b , where $b = t \times s$ is an index block, will denote the restriction of a matrix $A \in \mathbb{R}^{I \times J}$ to $\mathbb{R}^{t \times s}$. Although the above discretization methods have principle differences, they can be treated in a common way when approximating them by data-sparse methods. For $t \subset I$ and $s \subset J$ we introduce the following linear operators $\mathcal{P}_t : L^2(\Gamma) \rightarrow \mathbb{R}^t$, $\mathcal{Q}_s : L^2(\Gamma) \rightarrow \mathbb{R}^s$. For $i \in t$ and $j \in s$ let

$$\begin{aligned} (\mathcal{P}_t f)_i &= \int_{\Gamma} f(x) \psi_i(x) \, ds_x, & (\mathcal{Q}_s f)_j &= \int_{\Gamma} f(x) \varphi_j(x) \, ds_x && \text{Galerkin method} \\ (\mathcal{P}_t f)_i &= f(y_i), & (\mathcal{Q}_s f)_j &= \int_{\Gamma} f(x) \varphi_j(x) \, ds_x && \text{collocation method} \\ (\mathcal{P}_t f)_i &= f(y_i), & (\mathcal{Q}_s f)_j &= \omega_j f(y_j) && \text{Nyström method.} \end{aligned}$$

With this notation, each block A_{ts} of the stiffness matrix $A \in \mathbb{R}^{I \times J}$ takes the form

$$A_{ts} = \mathcal{P}_t \mathcal{K} \mathcal{Q}_s^*, \tag{1.2}$$

where $\mathcal{Q}_s^* : \mathbb{R}^s \rightarrow L^2(\Gamma)$ is the adjoint of \mathcal{Q}_s defined by

$$(\mathcal{Q}_s^* z, f)_{L^2(\Gamma)} = (z, \mathcal{Q}_s f) \quad \text{for all } z \in \mathbb{R}^s, f \in L^2(\Gamma). \tag{1.3}$$

Additionally, we define the supports

$$\text{supp } \mathcal{P} := \mathbb{R}^d \setminus D,$$

where D is the largest open set such that $\mathcal{P}\varphi = 0$ for all $\varphi \in L^2(\Gamma)$ satisfying $\text{supp } \varphi \subset D$. With this definition let

$$X_i := \text{supp } \mathcal{P}_i, \quad i \in I, \quad \text{and} \quad Y_j := \text{supp } \mathcal{Q}_j, \quad j \in J.$$

Data-sparse representations of the fully populated matrix A are often based on kernel approximations by degenerate kernels

$$\tilde{\kappa}(x, y) := \sum_{\ell=1}^k u_\ell(x)v_\ell(y) \tag{1.4}$$

on domains that are relatively to their diameters far enough away from each other. The corresponding block in the rows t and columns s of the dense matrix A will therefore be approximated by the matrix UV^T of rank at most k , where $U \in \mathbb{R}^{t \times k}$, $V \in \mathbb{R}^{s \times k}$ with entries

$$U_{i\ell} := \mathcal{P}_i u_\ell \quad \text{and} \quad V_{j\ell} := \mathcal{Q}_j v_\ell, \quad i \in t, \quad s \in j, \quad \ell = 1, \dots, k. \tag{1.5}$$

If k is small compared with the block sizes $|t|$ and $|s|$, the amount of storage for the block $t \times s$ and the cost for multiplying this block by a vector reduces from $|t| \cdot |s|$ to $k(|t| + |s|)$. On the other hand, κ is evaluated without approximation on domains which are close to each other. This approach is general in the sense that it can be applied to the singularity function $S(x, y)$ of any elliptic operator with constant coefficients; see Theorem 3.3. It relies on the fact that κ is singular for $x = y$ but smooth if $x \neq y$. Based on the above idea, fast summation methods provide an (approximate) dense matrix-vector multiplication of almost¹ linear complexity and can hence be used to accelerate any Krylov subspace method. Furthermore, the storage requirements of these methods is also almost linear. If the accuracy of the approximation lies within the range of other unavoidable errors like discretization errors, the solution will not notice this additional approximation.

The starting point of fast summation methods was the fast multipole method (FMM) published by Rokhlin [22], which concentrated on the fast multiplication of matrices of Nyström type by a vector. In the meanwhile variants of the FMM have been adapted to many different kernel functions; see for instance [21] for the application to elasticity. The panel clustering method proposed by Hackbusch and Nowak [14] was designed for the fast matrix-vector multiplication of collocation matrices; see [15] for the application to elastostatic problems.

The main difficulty of the FMM and the panel clustering method is that the functions u_i and v_i in (1.4) have to be known explicitly. Finding appropriate expansions can be quite cumbersome for instance for anisotropic elasticity. Furthermore, since k in (1.4) is chosen a priori so as to satisfy a prescribed accuracy in \mathbb{R}^d , k will usually be too large. Additionally, for boundary integral operators it is very likely that an approximation on a pair of domains within a common hyperplane is to be computed. In this case the problem is only $(d-1)$ -dimensional and k could be chosen much smaller. This led to algebraic methods like the mosaic skeleton method [27] and hierarchical (\mathcal{H} -) matrices [12, 13], which work on the matrix level and thus better adapt to the properties of the discrete operator.

An interesting observation due to Tyrtysnikov et al. [9] was the existence of low-rank approximants based on few of the original matrix entries. If such an approximant can be computed with reasonable effort, this is of great practical importance since existing computer codes can easily be modified, whereas the FMM and panel clustering methods require a complete recoding of the matrix-vector multiplication including computation of coefficients. Stimulated by this existence result, the adaptive cross approximation algorithm (ACA) [2, 3] for the computation of these low-rank approximants in the case of Nyström- and collocation matrices was published. This method is nowadays used in applications such as electromagnetics [18, 7] and in combination with algebraic multigrid [19].

¹a complexity that is linear up to logarithmic factors

Up to now, the proofs relied on the fact that collocation and Nyström matrices arise from evaluating functions at given points. Therefore, interpolation results were applied. For Galerkin matrices an analogous result was doubtful due to the variational character of the formulation.

The aim of this article is to prove that the ACA algorithm can also be applied to Galerkin matrices. Hence, it is possible to approximate A by a matrix $A_{\mathcal{H}}$ which needs $\mathcal{O}(N \log^d N)$ units of storage and can be multiplied by a vector with $\mathcal{O}(N \log^d N)$ complexity. The computation of $A_{\mathcal{H}}$ requires $\mathcal{O}(N \log^{2d-1} N)$ arithmetical operations. The structure we are using to approximate A are hierarchical matrices. Basically, these are matrices with a hierarchical block partitioning in which each block is a matrix of low rank. In Sect. 2 a more detailed description of \mathcal{H} -matrices will be given. The asymptotic smoothness of the singularity matrix of elliptic operators with constant coefficients is proved in Sect. 3. An important example are the Lamé equations (see Sect. 4), which will be used in this article to demonstrate the efficiency of ACA. The asymptotic smoothness of the singularity matrix is the only prerequisite of ACA. In Sect. 5 this algorithm is shown to converge for Galerkin matrices. The resulting approximant will be shown to be optimal in the sense that, up to constants, the remainder is smaller than the remainder resulting from any kernel approximation. Furthermore, we comment on implementation aspects of ACA for nonsmooth boundaries. In Sect. 6 numerical results from the application of ACA to Galerkin discretizations of problems from three-dimensional linear elasticity are presented.

2 Hierarchical matrices

This section gives a brief overview over the structure of \mathcal{H} -matrices originally introduced by Hackbusch et al. [12, 13]. The efficiency of \mathcal{H} -matrices is based on a hierarchical partition P of the set of matrix indices $I \times J$ such that each block $b = t \times s$, $t \subset I$, $s \subset J$, can be approximated by a matrix of low-rank; i.e.,

$$A_b \approx UV^T, \quad U \in \mathbb{R}^{t \times k}, \quad V \in \mathbb{R}^{s \times k},$$

where k is small compared with $|t|$ and $|s|$.

As mentioned in the introduction (see (1.5)), for the block A_{ts} a low-rank matrix approximant can be found whenever the generating kernel function κ can be approximated where κ is evaluated. The operators \mathcal{P}_i and \mathcal{Q}_j guarantee that for computing the entry A_{ij} the kernel function κ is evaluated on $X_i \times Y_j$. Hence, for the subblock A_{ts} we have to evaluate κ on $X_t := \bigcup_{i \in t} X_i$ and $Y_s := \bigcup_{j \in s} Y_j$. It will be seen in Sect. 5 that in the case of asymptotically smooth kernels κ such an approximation exists if for a given real number $0 < \eta < 1$ it holds that

$$\min\{\text{diam } Y_t, \text{diam } X_s\} < \eta \text{dist}(Y_t, X_s). \quad (2.1)$$

Note that for each $i \in I$ there will always be $j \in J$ such that $X_i \cap Y_j \neq \emptyset$. Condition (2.1) will hence be violated for at least one entry in each row. In order to guarantee that A_b with $(i, j) \in b$ has low rank, the dimensions of A_b therefore have to be small. An appropriate partition can be generated by recursive subdivision of $I \times J$; see [2]. The complexity of constructing P for quasi-uniform grids can be estimated as $\mathcal{O}(N \log N)$.

The set of \mathcal{H} -matrices for a given partition P with blockwise rank $k \in \mathbb{N}$ is defined as

$$\mathcal{H}(P, k) := \{A \in \mathbb{R}^{I \times J} : \text{rank } A_b \leq k \text{ for all } b \in P\}.$$

Exploiting the hierarchical structure, it can be shown that provided each block $t \times s$ has complexity $k(|t| + |s|)$ the complexity of the whole matrix will be of the order $kN \log N$; see [2]. Hence, both storing $A \in \mathcal{H}(P, k)$ and the computation of matrix-vector products have complexity $kN \log N$, which makes \mathcal{H} -matrices well-suited for iterative schemes such as Krylov subspace methods. Additionally, approximate versions of the usual matrix operations can be defined (see [10]) which have

logarithmic-linear complexity. Furthermore, the approximate arithmetic allows to define approximate LU decompositions, which can be used for preconditioning coefficient matrices arising from boundary element applications; see [5].

3 Asymptotic smoothness of singularity matrices

The boundary element method is usually applied to elliptic operators with constant coefficients, since the kernel function κ in (1.1) has to be known explicitly. In this section we will therefore lay theoretical ground to the efficient treatment of systems of partial differential operators

$$(\mathcal{L}u)_k = - \sum_{i,j=1}^d \sum_{\ell=1}^{\nu} \partial_i(c_{ij}^{k\ell} \partial_j) u_\ell, \quad k = 1, \dots, \nu, \quad (3.1)$$

with constant coefficients $c_{ij}^{k\ell}$ such that \mathcal{L} is elliptic; i.e., the Legendre-Hadamard condition (see [8]) holds

$$\sum_{i,j=1}^d \sum_{k,\ell=1}^{\nu} c_{ij}^{k\ell} v_i w_k v_j w_\ell \geq \lambda_{\mathcal{L}} |v|^2 |w|^2 \quad \text{for all } v \in \mathbb{R}^d, w \in \mathbb{R}^{\nu}, \quad (3.2)$$

where $|\cdot|$ denotes the Euclidean norm, $\lambda_{\mathcal{L}} > 0$ and

$$\max_{i,j,k,\ell} |c_{ij}^{k\ell}| \leq \Lambda_{\mathcal{L}}.$$

The aim of this section is to show that the entries of singularity matrices $S : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{\nu \times \nu}$ satisfying

$$\mathcal{L}_y S(x, y) = \delta(x - y) I_\nu$$

are asymptotically smooth.

Definition 3.1. *A function $\kappa : \Omega \times \mathbb{R}^d \rightarrow \mathbb{R}$ satisfying $\kappa(x, \cdot) \in C^\infty(\mathbb{R}^d \setminus \{x\})$ for all $x \in \Omega$ is called asymptotically smooth in Ω with respect to y if constants c and γ can be found such that for all $x \in \Omega$ and all $\alpha \in \mathbb{N}_0^d$*

$$|D_y^\alpha \kappa(x, y)| \leq c p! \gamma^p R^{-p} \sup_{z \in B_R(y)} |\kappa(x, z)| \quad \text{for all } y \in \mathbb{R}^d \setminus \{x\},$$

where $R = |x - y|/2$ and $p = |\alpha|$.

As usual we denote by D_y^α the partial derivative

$$D_y^\alpha = \left(\frac{\partial}{\partial y_1} \right)^{\alpha_1} \cdots \left(\frac{\partial}{\partial y_d} \right)^{\alpha_d}.$$

Asymptotical smoothness of the entries of S is a prerequisite of the ACA algorithm. It is usually checked for each singularity matrix under investigation. The following theory states that the entries of operators (3.1) will always have this property.

Let \mathcal{F} denote the Fourier transform which is assumed to be normalized such that $\mathcal{F}^* = \mathcal{F}^{-1}$. From $\mathcal{F}[D^\alpha u](\xi) = \xi^\alpha (2\pi i)^{|\alpha|} \mathcal{F}u(\xi)$, it follows that

$$-\partial_i c_{ij}^{k\ell} \partial_j u_k = -\mathcal{F}^* \mathcal{F}(\partial_i c_{ij}^{k\ell} \partial_j u_k) = (2\pi)^2 \mathcal{F}^*(c_{ij}^{k\ell} \xi_i \xi_j \mathcal{F}u_k)$$

and using (3.2) for $u \in [H_0^1(\Omega)]^\nu$ one has

$$\sum_{i,j=1}^d \sum_{k,\ell=1}^\nu \int_{\Omega} c_{ij}^{k\ell} \partial_i u_k \partial_j u_\ell \, dx = \int_{\Omega} (\mathcal{L}u, u) \, dx = (2\pi)^2 \sum_{i,j=1}^d \sum_{k,\ell=1}^\nu \int_{\Omega} c_{ij}^{k\ell} \xi_i \xi_j \mathcal{F}u_k \mathcal{F}u_\ell \, dx \quad (3.3a)$$

$$\geq (2\pi)^2 \lambda_{\mathcal{L}} \sum_{i=1}^d \sum_{\ell=1}^\nu \int_{\Omega} |\xi_i \mathcal{F}u_\ell|^2 \, dx = \lambda_{\mathcal{L}} \sum_{i=1}^d \sum_{\ell=1}^\nu \int_{\Omega} |\mathcal{F}(\partial_i u_\ell)|^2 \, dx \quad (3.3b)$$

$$= \lambda_{\mathcal{L}} \sum_{i=1}^d \sum_{\ell=1}^\nu \|\partial_i u_\ell\|_{L^2}^2 = \lambda_{\mathcal{L}} \|Du\|_{L^2}^2. \quad (3.3c)$$

The last estimate does not hold for variable coefficients $c_{ij}^{k\ell}$. In this case, either one has to consider operators $\mathcal{L} = -c_{ij}^{k\ell} \partial_i \partial_j$ instead of the divergence form (3.1) or the stronger condition

$$\sum_{i,j=1}^d \sum_{k,\ell=1}^\nu c_{ij}^{k\ell} v_{ik} v_{j\ell} \geq \lambda_{\mathcal{L}} \|V\|_F^2 \quad \text{for all } V \in \mathbb{R}^{d \times \nu}$$

has be imposed instead of (3.2). Here, $\|\cdot\|_F$ denotes the Frobenius norm.

In order to prove asymptotic smoothness, we first derive the following Caccioppoli-type inequality with the usual technique which uses cut-off functions.

Lemma 3.2. *Assume that $u \in [H^1(\Omega)]^\nu$ such that $\mathcal{L}u = 0$ in Ω . Then for any compact set $K \subset \Omega$ it holds that*

$$\|Du\|_{L^2(K)} \leq \frac{c_{\mathcal{L}}}{\sigma} \|u\|_{L^2(\Omega)},$$

where

$$c_{\mathcal{L}} = \sqrt{8d^2 \nu \frac{\Lambda_{\mathcal{L}}}{\lambda_{\mathcal{L}}} \left(1 + 8d\nu \frac{\Lambda_{\mathcal{L}}}{\lambda_{\mathcal{L}}}\right)} \quad \text{and} \quad \sigma = \text{dist}(K, \partial\Omega).$$

Proof. Let $\eta \in C^1(\Omega)$ satisfy $0 \leq \eta \leq 1$, $\eta = 1$ in K , $\eta = 0$ in a neighborhood of $\partial\Omega$ and $|\partial_i \eta| \leq 2/\sigma$, $i = 1, \dots, d$, in Ω . With $u \in [H^1(\Omega)]^\nu$ it follows that $\eta^2 u \in [H_0^1(\Omega)]^\nu$. From

$$\partial_i(\eta^2 u_k) \partial_j u_\ell = \partial_i(\eta u_k) \partial_j(\eta u_\ell) + u_k \partial_i \eta \partial_j(\eta u_\ell) - u_\ell \partial_i(\eta u_k) \partial_j \eta - u_k u_\ell \partial_i \eta \partial_j \eta$$

we obtain using $\int_{\Omega} (\mathcal{L}u, \eta^2 u) \, dx = 0$ that for any $\epsilon > 0$

$$\begin{aligned} & \sum_{i,j=1}^d \sum_{k,\ell=1}^\nu \int_{\Omega} c_{ij}^{k\ell} \partial_i(\eta u_k) \partial_j(\eta u_\ell) \, dx \\ & \leq \Lambda_{\mathcal{L}} \sum_{i,j=1}^d \sum_{k,\ell=1}^\nu \int_{\Omega} |u_k| |u_\ell| |\partial_i \eta| |\partial_j \eta| + 2|u_k| |\partial_i \eta| |\partial_j(\eta u_\ell)| \, dx \\ & \leq 4\Lambda_{\mathcal{L}} \frac{d}{\sigma} \sum_{j=1}^d \sum_{k,\ell=1}^\nu \int_{\Omega} \frac{1}{\sigma} |u_k| |u_\ell| + |u_k| |\partial_j(\eta u_\ell)| \, dx \\ & \leq 4\Lambda_{\mathcal{L}} \frac{d}{\sigma} \left(\nu \frac{d}{\sigma} \|u\|_{L^2}^2 + \sum_{j=1}^d \sum_{k,\ell=1}^\nu \left(\int_{\Omega} |u_k|^2 \, dx \right)^{1/2} \left(\int_{\Omega} |\partial_j(\eta u_\ell)|^2 \, dx \right)^{1/2} \right) \\ & \leq 4\Lambda_{\mathcal{L}} \frac{d\nu}{\sigma} \left(\frac{d}{\sigma} \|u\|_{L^2}^2 + \frac{d}{\epsilon} \sum_{k=1}^\nu \int_{\Omega} |u_k|^2 \, dx + \epsilon \sum_{j=1}^d \sum_{\ell=1}^\nu \int_{\Omega} |\partial_j(\eta u_\ell)|^2 \, dx \right) \\ & = 4\Lambda_{\mathcal{L}} \frac{d^2 \nu}{\sigma^2} \left(1 + \frac{\sigma}{\epsilon}\right) \|u\|_{L^2(\Omega)}^2 + 4\Lambda_{\mathcal{L}} \frac{d\nu}{\sigma} \epsilon \int_{\Omega} \|D(\eta u)\|_F^2 \, dx. \end{aligned}$$

Using (3.3), we obtain

$$\lambda_{\mathcal{L}} \int_{\Omega} \|D(\eta u)\|_F^2 dx \leq 4\Lambda_{\mathcal{L}} \frac{d^2\nu}{\sigma^2} \left(1 + \frac{\sigma}{\epsilon}\right) \|u\|_{L^2(\Omega)}^2 + 4\Lambda_{\mathcal{L}} \frac{d\nu}{\sigma} \epsilon \int_{\Omega} \|D(\eta u)\|_F^2 dx.$$

This leads to

$$\|Du\|_{L^2(K)}^2 \leq \|D(\eta u)\|_{L^2(\Omega)}^2 \leq \frac{1}{\sigma^2} \frac{4\Lambda_{\mathcal{L}} d^2\nu (1 + \sigma/\epsilon)}{\lambda_{\mathcal{L}} - 4\Lambda_{\mathcal{L}} d\nu\epsilon/\sigma} \|u\|_{L^2(\Omega)}^2.$$

Choosing $\epsilon = \lambda_{\mathcal{L}}\sigma/(8\Lambda_{\mathcal{L}}d\nu)$ gives the desired result. \square

The Lamé equations, which arise from linear isotropic elasticity, read

$$-\mu\vec{\Delta}u - (\lambda + \mu)\nabla\operatorname{div}u = f.$$

Here, λ and μ denote the Lamé constants. The last operator is in the class of operators (3.1) with coefficients $c_{ij}^{k\ell} = \mu\delta_{ij}\delta_{k\ell} + (\lambda + \mu)\delta_{ik}\delta_{j\ell}$ and $\nu = d$. Hence,

$$\sum_{i,j,k,\ell=1}^d c_{ij}^{k\ell} v_i w_k v_j w_\ell = \mu|v|^2|w|^2 + (\lambda + \mu)(v \cdot w)^2 \geq \mu|v|^2|w|^2$$

and the Lamé operator satisfies condition (3.2). Another example for systems of partial differential operators is the equation $\mathcal{L}u = f$ with the operator

$$\mathcal{L} := \operatorname{curl}\operatorname{curl} - \alpha\nabla\operatorname{div}, \quad \alpha > 0,$$

which arises from Maxwell's equations. Since $\operatorname{curl}\operatorname{curl} = -\vec{\Delta} + \nabla\operatorname{div}$, we obtain that

$$\mathcal{L} = -\vec{\Delta} + (1 - \alpha)\nabla\operatorname{div}.$$

Hence, the *curl-curl operator* has a similar structure as the Lamé operator. However, it differs by the sign of the $\nabla\operatorname{div}$ part. As a consequence, $c_{ij}^{k\ell} = \delta_{ij}\delta_{k\ell} + (\alpha - 1)\delta_{ik}\delta_{j\ell}$ gives

$$\sum_{i,j,k,\ell=1}^d c_{ij}^{k\ell} v_i w_k v_j w_\ell = |v|^2|w|^2 + (\alpha - 1)(v \cdot w)^2.$$

If $\alpha \geq 1$, then the last expression is obviously bounded from below by $|v|^2|w|^2$. In the other case, $0 < \alpha < 1$, we observe that

$$|v|^2|w|^2 + (\alpha - 1)(v \cdot w)^2 = \alpha|v|^2|w|^2 + (1 - \alpha)[|v|^2|w|^2 - (v \cdot w)^2] \geq \alpha|v|^2|w|^2$$

due to the Cauchy-Schwarz inequality. Hence, also the curl-curl operator satisfies (3.2).

In order to derive pointwise estimates, we need the following estimate for u satisfying $\mathcal{L}u = 0$ in $B_r(x) \subset \mathbb{R}^d$ and $k \in \mathbb{N}$

$$\|u\|_{H^k(B_\rho(x))} \leq c(k, \rho, r) \|u\|_{L^2(B_r(x))} \quad \text{for all } 0 < \rho < r, \quad (3.4)$$

where c depends on the coefficients of \mathcal{L} . Estimate (3.4) can be derived by applying Lemma 3.2 iteratively on a nested sequence of balls $B_{r_\ell}(x)$, $\ell = 1, \dots, k$. Due to the Sobolev embedding theorem, estimate (3.4) states that \mathcal{L} -harmonic functions are locally C^∞ . In particular, (3.4) gives for the choice $k = d + 1$

$$\sup_{B_\rho(x)} |u| \leq c \|u\|_{H^{d+1}(B_\rho(x))} \leq \tilde{c}(\rho, r) \|u\|_{L^2(B_r(x))}.$$

Using a rescaling argument, we obtain for $x \in \Omega$ and $0 \leq r \leq \text{dist}(x, \partial\Omega)$

$$\sup_{B_\rho(x)} |u| \leq c_R r^{-d/2} \|u\|_{L^2(B_r(x))}, \quad 0 < \rho < r, \quad (3.5)$$

with $c_R > 0$ independent of ρ and r .

The next lemma shows that the entries S_{ij} , $i, j = 1, \dots, \nu$, of the singularity matrix S are asymptotically smooth. We already know that $S_{ij}(x, \cdot) \in C^\infty(\Omega)$, $i, j = 1, \dots, \nu$, is \mathcal{L} -harmonic in each $\Omega \subset \mathbb{R}^d \setminus \{x\}$.

Theorem 3.3. *The entries of the singularity matrix S of \mathcal{L} are asymptotically smooth in \mathbb{R}^d with respect to y .*

Proof. Let $x \in \mathbb{R}^d$ be arbitrary but fixed. For $y \in \mathbb{R}^d \setminus \{x\}$ let $R = |x - y|/2$. Assume that a function u is \mathcal{L} -harmonic in $B_r(y)$, $0 < r < R$. Choosing $0 < \rho < r$ and $\rho' := (r + \rho)/2$, we obtain from (3.5) and Lemma 3.2 that

$$\sup_{z \in B_\rho(y)} |\partial_{z_i} u(z)|^2 \leq \frac{c_R^2}{\rho'^d} \int_{B_{\rho'}(y)} |\partial_{z_i} u(z)|^2 dz \leq \frac{c_R^2 c_{\mathcal{L}}^2}{\rho'^d (r - \rho')^2} \int_{B_r(y)} |u(z)|^2 dz \quad (3.6a)$$

$$\leq 2^{d+2} \frac{\omega_d r^d}{(r + \rho)^d} \frac{c_R^2 c_{\mathcal{L}}^2}{(r - \rho)^2} \sup_{z \in B_r(y)} |u(z)|^2. \quad (3.6b)$$

Here, ω_d denotes the volume of the unit ball in \mathbb{R}^d .

Let $\alpha \in \mathbb{N}_0^d$ be a multi-index and $p = |\alpha|$. We define a nested sequence of balls

$$B_k = \{z \in \mathbb{R}^d : |z - y| < Rk/(p + 1)\}, \quad k = 1, \dots, p + 1,$$

centered at y . Then $B_k \subset B_R(y) \subset \mathbb{R}^d \setminus \{x\}$ and $\text{dist}(B_k, \partial B_{k+1}) = R/(p + 1)$. Estimate (3.6) for $\rho = Rk/(p + 1)$ and $r = R(k + 1)/(p + 1)$ reads

$$\begin{aligned} \sup_{z \in B_k} |\partial_{z_i} u(z)| &\leq 2^{d/2+1} \left(\frac{k + 1}{2k + 1} \right)^{d/2} \frac{c_R c_{\mathcal{L}} \sqrt{\omega_d}}{R} (p + 1) \sup_{z \in B_{k+1}} |u(z)| \\ &= c'_{\mathcal{L}} \frac{p + 1}{R} \left(\frac{2k + 2}{2k + 1} \right)^{d/2} \sup_{z \in B_{k+1}} |u(z)|, \quad k = 1, \dots, p, \end{aligned}$$

where $c'_{\mathcal{L}} := 2c_R c_{\mathcal{L}} \sqrt{\omega_d}$. Applying the last estimate consecutively to the p partial derivatives of the function $S_{ij}(x, \cdot)$, which together with its derivatives $\partial_y^\alpha S_{ij}(x, y)$ is \mathcal{L} -harmonic in $B_R(y) \subset \mathbb{R}^d$ for arbitrary $\alpha \in \mathbb{N}_0^d$, we end up with

$$\sup_{z \in B_1} |D_z^\alpha S_{ij}(x, z)| \leq (p + 1)^{d/4} \left(\frac{c'_{\mathcal{L}} (p + 1)}{R} \right)^p \sup_{z \in B_{p+1}} |S_{ij}(x, z)|$$

since $\prod_{k=1}^p \frac{2k+2}{2k+1} \leq \sqrt{p+1}$. Using $(p + 1)^p \leq ep^p$ and Stirling's approximation

$$\sqrt{2\pi p} \left(\frac{p}{e} \right)^p < p!,$$

we obtain

$$|D_y^\alpha S_{ij}(x, y)| \leq \frac{e}{\sqrt{2\pi}} 2^{d/4} p! \left(2^{\frac{d-2}{8}} \frac{c'_{\mathcal{L}} e}{R} \right)^p \sup_{z \in B_R(y)} |S_{ij}(x, z)|$$

due to $p \leq 2^{p/2}$. □

Note that neither the smoothness of $S(x, y)$ with respect to x nor smoothness properties of the surface Γ are necessary. As an example, any partial derivative $\partial_x^\alpha S(x, y)$ is asymptotically smooth with respect to y . The hyper-singular operator, the kernel of which contains derivatives with respect to both variables, is usually expressed in terms of the single layer operator.

Although we could apply ACA directly to the arising Galerkin matrix, in the case of discretizations of the Lamé equations it is more efficient to exploit the following representation.

4 Galerkin discretization of the Lamé equations

We consider an elastic body $\Omega \subset \mathbb{R}^3$ with a Lipschitz boundary Γ . Suppose that a displacement of the body g_D is given on $\Gamma_D \subset \Gamma$ and some force g_N is applied to $\Gamma_N \subset \Gamma = \overline{\Gamma}_D \cup \overline{\Gamma}_N$. To find displacements and forces inside Ω , we set up a mixed boundary value problem for the Lamé system:

$$-\mu \vec{\Delta} u - (\lambda + \mu) \nabla \operatorname{div} u = 0, \quad x \in \Omega, \quad (4.1a)$$

$$u = g_D, \quad x \in \Gamma_D, \quad (4.1b)$$

$$t = Tu = g_N, \quad x \in \Gamma_N. \quad (4.1c)$$

Here, $u : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the displacement vector,

$$T_x u(x) = \lambda \operatorname{div} u(x) \cdot n(x) + 2\mu \frac{\partial}{\partial n} u(x) + \mu n(x) \times \operatorname{curl} u(x) \quad (4.2)$$

is the traction on the boundary. Lamé constants λ and μ relate to the elasticity module E and the Poisson ratio ν of the material as follows

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}. \quad (4.3)$$

The problem (4.1) admits the following symmetric boundary integral formulation (for details we refer to Sirtori [25] and Steinbach [26]):

$$(V\tilde{t})(x) - (K\tilde{u})(x) = \left(\frac{1}{2}I + K\right) \tilde{g}_D(x) - (V\tilde{g}_N)(x) \quad \text{for } x \in \Gamma_D, \quad (4.4a)$$

$$(K'\tilde{t})(x) + (D\tilde{u})(x) = \left(\frac{1}{2}I - K'\right) \tilde{g}_N(x) - (D\tilde{g}_D)(x) \quad \text{for } x \in \Gamma_N, \quad (4.4b)$$

where $\tilde{u} = u - \tilde{g}_D$, $\tilde{t} = t - \tilde{g}_N$, \tilde{g}_N and \tilde{g}_D are extensions of g_N and g_D to the whole boundary Γ . Here

$$(Vt)(x) = \int_{\Gamma} [U^*(x, y)]^T t(y) \, ds_y,$$

$$(Ku)(x) = \int_{\Gamma} [T_y^* U^*(x, y)]^T u(y) \, ds_y,$$

$$(K't)(x) = \int_{\Gamma} [T_x U^*(x, y)]^T t(y) \, ds_y,$$

$$(Du)(x) = -T_x \int_{\Gamma} [T_y^* U^*(x, y)]^T u(y) \, ds_y,$$

are the single layer, double layer, adjoint double layer potentials, and a hyper-singular operator respectively.

$$U_{k\ell}^*(x, y) = \frac{1+\nu}{8\pi E(1-\nu)} \left[(3-4\nu) \frac{\delta_{k\ell}}{|x-y|} + \frac{(x_k - y_k)(x_\ell - y_\ell)}{|x-y|^3} \right]$$

is the fundamental solution. As it was shown in [17] (see also [26]), the action of K and K' can be expressed in terms of V , V_L , K_L , and R

$$\begin{aligned}\langle Ku, t \rangle &= 2\mu \langle VRu, t \rangle + \langle K_L u, t \rangle - \langle V_L Ru, t \rangle \\ \langle K't, v \rangle &= 2\mu \langle Vt, Rv \rangle + \langle K'_L t, v \rangle - \langle V_L t, Rv \rangle,\end{aligned}\tag{4.5}$$

where V_L , K_L , and K'_L are the corresponding operators for the Laplace equation:

$$\begin{aligned}V_L u &= \frac{1}{4\pi} \int_{\Gamma} \frac{1}{|x-y|} u(y) \, ds_y \\ K_L u &= \frac{1}{4\pi} \int_{\Gamma} \left[\frac{\partial}{\partial n_y} \frac{1}{|x-y|} \right] u(y) \, ds_y\end{aligned}$$

and

$$R_{ij} = n_j \frac{\partial}{\partial x_i} - n_i \frac{\partial}{\partial x_j}.$$

The action of the hyper-singular operator can also be expressed in terms of V , V_L , and R (see [17], [26]):

$$\begin{aligned}\langle Du, v \rangle &= \int_{\Gamma} \int_{\Gamma} \frac{\mu}{4\pi |x-y|} \left(\sum_{k=1}^3 \frac{\partial}{\partial S_k} u(y) \frac{\partial}{\partial S_k} v(x) \right) + \\ &+ (Rv)^T(x) \left(\frac{\mu}{2\pi |x-y|} I - 4\mu^2 U^*(x, y) \right) (Ru)(y) + \\ &+ \sum_{i,j,k=1}^3 (R_{kj} v_i)(x) \frac{1}{|x-y|} (R_{ki} u_j)(y) \, ds_x \, ds_y.\end{aligned}\tag{4.6}$$

Here,

$$\frac{\partial}{\partial S_1} = R_{32}, \quad \frac{\partial}{\partial S_2} = R_{13}, \quad \frac{\partial}{\partial S_3} = R_{21}.$$

To see how useful the above relations are, we proceed with the variational formulation of the problem.

Consider Sobolev spaces $\tilde{H}^s(\Gamma_L) = \{v = \tilde{v}|_{\Gamma_L} : \tilde{v} \in H^s(\Gamma), \text{supp } \tilde{v} \subset \Gamma_L\}$, $L \in \{N, D\}$. We multiply equations (4.4a) by test functions $\tau \in \tilde{H}^{-1/2}(\Gamma_D)$ and $v \in \tilde{H}^{1/2}(\Gamma_N)$ respectively and integrate the resulting equations over the surface. This gives

$$a(\tilde{u}, \tilde{t}; v, \tau) = f(v, \tau) \quad \text{for all } (v, \tau) \in \tilde{H}^{1/2}(\Gamma_N) \times \tilde{H}^{-1/2}(\Gamma_D),$$

where

$$a(\tilde{u}, \tilde{t}; v, \tau) = \langle V\tilde{t}, \tau \rangle_{L^2(\Gamma_D)} - \langle K\tilde{t}, \tau \rangle_{L^2(\Gamma_D)} + \langle K'\tilde{t}, v \rangle_{L^2(\Gamma_N)} + \langle D\tilde{u}, v \rangle_{L^2(\Gamma_N)}$$

and

$$f(v, \tau) = \left\langle \left(\frac{1}{2}I + K \right) \tilde{g}_D, \tau \right\rangle_{L^2(\Gamma_D)} - \langle V\tilde{g}_N, \tau \rangle_{L^2(\Gamma_D)} + \left\langle \left(\frac{1}{2}I - K' \right) \tilde{g}_N, v \right\rangle_{L^2(\Gamma_N)} - \langle D\tilde{g}_D, v \rangle_{L^2(\Gamma_N)}.$$

We discretize spaces $H^{-1/2}(\Gamma) \approx \text{span}(\{\varphi_i\}_{i=1}^{N'})$ and $H^{1/2}(\Gamma) \approx \text{span}(\{\psi_j\}_{j=1}^{N'})$ looking for the solution in the form

$$\tilde{u}_h(x) = \sum_{j=1}^{N'} \begin{bmatrix} u_{x,j} \\ u_{y,j} \\ u_{z,j} \end{bmatrix} \psi_j(x), \quad \tilde{t}_h(x) = \sum_{i=1}^N \begin{bmatrix} t_{x,i} \\ t_{y,i} \\ t_{z,i} \end{bmatrix} \varphi_i(x).\tag{4.7}$$

Since \tilde{u}_h and \tilde{t}_h represent the difference between the unknown functions and an extension of the given data, some coefficients in the above sums are known. Substituting (4.7) into the variational formulation gives the following system of linear algebraic equations for the partially unknown coefficient vectors $u_h = \{u_{x,j}, u_{y,j}, u_{z,j}\}_{j=1}^{N'}$ and $t_h = \{t_{x,i}, t_{y,i}, t_{z,i}\}_{i=1}^N$

$$\begin{bmatrix} V_h & -K_h \\ K_h^T & D_h \end{bmatrix} \begin{bmatrix} t_h \\ u_h \end{bmatrix} = \begin{bmatrix} -V_h & \frac{1}{2}M_h + K_h \\ \frac{1}{2}M_h - K_h^T & -D_h \end{bmatrix} \begin{bmatrix} \tilde{g}_N \\ \tilde{g}_D \end{bmatrix} =: \begin{bmatrix} f_N \\ f_D \end{bmatrix}. \quad (4.8)$$

The elements of the above matrices are

$$(V_h)_{kl} = \langle V\varphi_l, \varphi_k \rangle_{L^2(\Gamma_D)}, \quad (K_h)_{kj} = \langle K\psi_j, \varphi_k \rangle_{L^2(\Gamma_D)}, \quad (D_h)_{ij} = \langle D\psi_j, \psi_i \rangle_{L^2(\Gamma_N)}.$$

To deal with a large number of unknown coefficients, the solution of the discrete system (4.8) is performed using an iterative scheme. Since V is elliptic, our symmetric Galerkin matrix V_h is positive definite. We multiply the first equation in (4.8) by $-K_h^T V_h^{-1}$ and add the second equation to the result obtaining an equation for \tilde{u}_h

$$(D_h + K_h^T V_h^{-1} K_h) \tilde{u}_h = f_D - K_h^T V_h^{-1} f_N.$$

The Schur complement $D_h + K_h^T V_h^{-1} K_h$ in the above equation is symmetric and positive definite (see [26]), therefore a conjugate gradient scheme can be applied to find \tilde{u}_h . Then t_h can be found from the first equation of (4.8).

To use the above solution method, we only need to implement matrix-vector multiplication routines. We do it using representations (4.5) and (4.6), avoiding explicit computations of the entries of D_h and K_h . The only dense matrices that need to be stored are V_h , $(V_L)_h$, and $(K_L)_h$. In what follows, we show that admissible blocks of these matrices can be efficiently approximated using low-rank matrices by means of the ACA method.

5 Adaptive Cross Approximation

In this section we present a method which generates an \mathcal{H} -matrix approximant of the Galerkin matrix (1.2). In contrast to other methods like fast multipole, panel clustering, etc., the low-rank approximant is not generated by replacing the kernel function of the integral operator. The algorithm uses few of the original matrix entries to compute the low-rank matrix. Note that this does not require to build the whole matrix beforehand. The proposed algorithm will specify which entries have to be computed. Obviously, this has the advantage that the algorithm can be applied to any appropriate problem without changing the algorithm, since only the original entries change, and, what is even more important for practice, existing codes for the computation of the entries can be used, whereas methods like multipole require a complete recoding.

The singular value decomposition would find the lowest rank that is required for a given accuracy; cf. [16]. However, its computational complexity makes it unattractive for large-scale computations. The presented technique can be regarded as an efficient replacement which is tailored to asymptotically smooth kernels. Note that not the kernel function itself but only the information that the kernel is in this class of functions is required. This enables the design of a black-box algorithm for discrete integral operators with asymptotically smooth kernels.

We assume that a partition P has been generated as in Sect. 2. If a block $b \in P$ does not satisfy (2.1), then its entries are stored without approximation. Therefore, this case is not treated here. If condition (2.1) holds for b , then we will show how to construct a low-rank approximant to the original matrix entries $A \in \mathbb{R}^{t \times s}$ corresponding to $b = t \times s$. For simplicity, in the rest of this section we will therefore focus on a single block $A \in \mathbb{R}^{m \times n}$.

The idea of the algorithm is as follows. Starting from $R_0 := A$, find a nonzero pivot in R_k , say (i_k, j_k) , and subtract a scaled outer product of the i_k -th row and the j_k -th column:

$$R_{k+1} := R_k - [(R_k)_{i_k j_k}]^{-1} (R_k)_{1:m, j_k} (R_k)_{i_k, 1:n}, \quad (5.1)$$

where we use the notations $(R_k)_{i, 1:n}$ and $(R_k)_{1:m, j}$ for the i -th row and the j -th column of R_k , respectively. It will turn out that j_k should be chosen the maximum element in modulus of the i_k -th row of R_{k-1} ; i.e.,

$$|(R_k)_{i_k j_k}| = \max_{j=1, \dots, n} |(R_k)_{i_k j}|. \quad (5.2)$$

The choice of i_k will be treated in Sect. 5.2.

Since in the k -th step only the entries in the j_k -th column and the i_k -th row of R_k are used to compute R_{k+1} , there is no need to build the whole matrix R_k . Taking advantage of this, the following algorithm is an efficient reformulation of (5.1). Note that the vectors u_k and \tilde{v}_k coincide with $(R_k)_{1:m, j_k}$ and $(R_k)_{i_k, 1:n}^T$, respectively.

```

Let  $k = 1$ ;  $Z = \emptyset$ ;
repeat
  find  $i_k$  as described in Sect. 5.2
   $\tilde{v}_k := a_{i_k, 1:n}$ 
  for  $\ell = 1, \dots, k - 1$  do  $\tilde{v}_k := \tilde{v}_k - (u_\ell)_{i_k} v_\ell$ 
   $Z := Z \cup \{i_k\}$ 
  if  $\tilde{v}_k$  does not vanish then
     $j_k := \operatorname{argmax}_{j=1, \dots, n} |(\tilde{v}_k)_j|$ ;  $v_k := (\tilde{v}_k)_{j_k}^{-1} \tilde{v}_k$ 
     $u_k := a_{1:m, j_k}$ 
    for  $\ell = 1, \dots, k - 1$  do  $u_k := u_k - (v_\ell)_{j_k} u_\ell$ .
     $k := k + 1$ 
  endif
until the stopping criterion (5.3) is fulfilled or  $Z = \{1, \dots, m\}$ 

```

Algorithm 5.1: Adaptive Cross Approximation (ACA)

In the set Z the vanishing rows of the R_k 's are collected. If the i_k -th row of R_k is nonzero and hence is used as v_k , it is also added to Z since the i_k -th row of the following remainder R_{k+1} will vanish. The matrix $S_k := \sum_{\ell=1}^k u_\ell v_\ell^T$ will be used as an approximation of $A = S_k + R_k$. Obviously, the rank of S_k is bounded by k .

Let $\varepsilon > 0$ be given. The following condition on k

$$\|u_{k+1}\|_2 \|v_{k+1}\|_2 \leq \frac{\varepsilon(1-\eta)}{1+\varepsilon} \|S_k\|_F \quad (5.3)$$

can be used as a stopping criterion; see [4]. The Frobenius norm $\|S_k\|_F$ of S_k can be computed with $\mathcal{O}(k^2(m+n))$ complexity. Therefore, the amount of numerical work required by Algorithm 5.1 is of the order $|Z|^2(m+n)$.

5.1 Error analysis

In order to estimate the efficiency of ACA, we have to find a bound for the norm of the remainder R_k . The analysis in the case of the Nyström method was done in [2], collocation matrices were treated in [3]. The proofs were quite technical. Therefore, in the following section we will present a short unified proof which also covers the case of Galerkin matrices. In contrast to Nyström and collocation matrices, which arise from evaluating functions at given points such that interpolation arguments can be applied, a vanishing row in a Galerkin matrix means that a function vanishes only in a weak sense.

Without loss of generality, in the rest of this section we assume that for the pivotal indices i_ℓ and j_ℓ it holds that $i_\ell = j_\ell = \ell$, $\ell = 1, \dots, k$. Then A has the decomposition

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{R}^{k \times k}, \quad (5.4)$$

where only the matrix blocks A_{11} , A_{12} and A_{21} have been used by Algorithm 5.1. Note that the (large) block $A_{22} \in \mathbb{R}^{(m-k) \times (n-k)}$ has never been touched. Since the determinant of A_{11} is the product of the pivots, A_{11} is invertible and we can express the remainder R_k of the approximation in terms of the original matrix A .

Lemma 5.1. *For R_k it holds that*

$$R_k = A - \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} A_{11}^{-1} \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & C_k \end{bmatrix},$$

where $C_k := A_{22} - A_{21}A_{11}^{-1}A_{12}$ is the Schur complement of A_{11} in A .

Proof. The assertion is obvious for $k = 1$. Assume that it holds for k and let A be decomposed in the following way

$$A = \begin{bmatrix} A_{11} & w & B \\ v^T & \alpha & y^T \\ C & x & D \end{bmatrix}, \quad A_{11} \in \mathbb{R}^{k \times k},$$

with vectors $x \in \mathbb{R}^{m-k-1}$, $y \in \mathbb{R}^{n-k-1}$, $v, w \in \mathbb{R}^k$, and $\alpha \in \mathbb{R}$. From (5.1) we see that

$$R_{k+1} = A - \begin{bmatrix} A_{11} & w \\ v^T & \alpha \\ C & x \end{bmatrix} \begin{bmatrix} A_{11}^{-1} + \gamma A_{11}^{-1} w v^T A_{11}^{-1} & -\gamma A_{11}^{-1} w \\ -\gamma v^T A_{11}^{-1} & \gamma \end{bmatrix} \begin{bmatrix} A_{11} & w & B \\ v^T & \alpha & y^T \end{bmatrix},$$

where $\gamma = (\alpha - v^T A_{11}^{-1} w)^{-1}$ and $\alpha - v^T A_{11}^{-1} w$ is the pivot, which is chosen nonzero. Since

$$\begin{bmatrix} A_{11} & w \\ v^T & \alpha \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} + \gamma A_{11}^{-1} w v^T A_{11}^{-1} & -\gamma A_{11}^{-1} w \\ -\gamma v^T A_{11}^{-1} & \gamma \end{bmatrix},$$

we obtain the desired result. \square

From Algorithm 5.1 it can be seen that the number $k^* := |Z|$ of zero columns in R_k may be larger than k ; i.e., we have $k^* \geq k$. This happens if in the ℓ -th step the algorithm comes across a zero vector \tilde{v}_ℓ and has to continue with another row. Without loss of generality, we assume that the indices corresponding to a zero row met in Algorithm 5.1 are $\{k+1, \dots, k^*\}$. Let A_{21} and A_{22} from (5.4) be decomposed in the following way

$$A_{21} = \begin{bmatrix} \hat{A}_{21} \\ \tilde{A}_{21} \end{bmatrix}, \quad A_{22} = \begin{bmatrix} \hat{A}_{22} \\ \tilde{A}_{22} \end{bmatrix}, \quad \hat{A}_{21} \in \mathbb{R}^{(k^*-k) \times k}, \quad \hat{A}_{22} \in \mathbb{R}^{(k^*-k) \times (n-k)}.$$

It is remarkable that although the approximant S_k has rank at most k , not k but k^* will determine the accuracy of the approximation as can be seen from the following lemma, which will be used to estimate the norm of the Schur complement and hence the norm of the remainder R_k . This observation also is of practical importance. Since a zero row is not lost for the approximation accuracy, the problem of finding nonzero pivots is save from leading to the computation of the whole matrix.

Lemma 5.2. *Let $X \in \mathbb{R}^{(m-k) \times k^*}$ be arbitrary, then*

$$C_k = \left\{ A_{22} - X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix} \right\} - \left\{ A_{21} - X \begin{bmatrix} A_{11} \\ \hat{A}_{21} \end{bmatrix} \right\} A_{11}^{-1} A_{12}. \quad (5.5)$$

Proof. It is easy to check that a zero row in R_ℓ will remain zero in R_k , $k \geq \ell$. Hence, $(R_k)_{i,1:n} = 0$, $i = k + 1, \dots, k^*$. From Lemma 5.1 it follows that

$$\hat{A}_{22} = \hat{A}_{21} A_{11}^{-1} A_{12}.$$

Adding and subtracting

$$X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix} = X \begin{bmatrix} A_{11} \\ \hat{A}_{21} \end{bmatrix} A_{11}^{-1} A_{12}$$

to and from C_k , we end up with what we were after. \square

The expressions from (5.5) appearing in curly braces will be estimated by a relation to the approximation of functions. What remains for an estimate of C_k , is a bound on the size of the coefficients $(A_{11}^{-1} A_{12})_{ij}$. By Cramer's rule it holds that

$$(A_{11}^{-1} A_{12})_{ij} = \frac{\det(a_1, \dots, a_{i-1}, a'_j, a_{i+1}, \dots, a_k)}{\det A_{11}},$$

where a_ℓ , $\ell = 1, \dots, k$, are the columns of A_{11} and a'_j is the j -th column of A_{12} . The coefficients $(A_{11}^{-1} A_{12})_{ij}$ in (5.5) should be as small as possible for C_k being small. The optimal choice of the pivots would be a submatrix A_{11} having maximum determinant in modulus. The method of pseudo-skeletons (cf. [28]) is based on this pivoting strategy. To find such a submatrix in A with reasonable effort however seems to be impossible. The pivoting strategy used in Algorithm 5.1 gives the following bound on the size of the entries in $A_{11}^{-1} A_{12}$. The proof can be done analogously to the proof of a similar result published in [2].

Lemma 5.3. *Assume that in each step we choose j_k so that (5.2) is satisfied. Then for $i = 1, \dots, k$ and $j = 1, \dots, n - k$ it holds that*

$$|\det(a_1, \dots, a_{i-1}, a'_j, a_{i+1}, \dots, a_k)| \leq 2^{k-i} |\det A_{11}|.$$

We are now ready to estimate the remainder R_k . For this purpose, the entries of R_k will be estimated by the approximation error

$$E_{ts}^\Xi := \max_{j \in s} \inf_{v \in \text{span } \Xi} \|\mathcal{K} Q_j^* - v\|_{\infty, Y_t} \quad (5.6)$$

in an arbitrary system of functions $\Xi := \{\Xi_1, \dots, \Xi_{k^*}\}$ with $\Xi_1 = 1$. The operator Q_j^* was previously defined in (1.3). If $k^* = \binom{p+d}{p} \sim p^d$ and $\Xi = \Pi_p^d$ is chosen to be algebraic polynomials, then it can be shown (see [23]) that

$$E_{ts}^\Xi \sim \eta^p$$

provided $t \times s$ satisfies (2.1). Note that $\mathcal{K} Q_j^*$ is asymptotically smooth with respect to y , since Q_j acts on the variable x only. Polynomials are not the only system of functions one can think of when approximation asymptotically smooth functions. For the approximation of the kernel $|x - y|^{-1}$ in \mathbb{R}^3 by spherical harmonics, the number of required functions to guarantee an approximation error of order η^p is of the order p^2 in contrast to p^3 when polynomials are used.

Assume that k steps of Algorithm 5.1 have been carried out and let $k^* = |Z|$ be the number of zero rows met. Before we consider Galerkin matrices in Theorem 5.6, the approximation error

associated with Algorithm 5.1 applied to collocation and Nyström matrices, i.e. the case $\mathcal{P}_i f = f(y_i)$, is estimated.

For the following theorem we assume the unisolvency of the system Ξ in the nodes y_i , $i = 1, \dots, k^*$; i.e.,

$$\det W \neq 0, \quad (5.7)$$

where $W \in \mathbb{R}^{k^* \times k^*}$ has the entries $w_{ij} = \Xi_j(y_i)$, $i, j = 1, \dots, k^*$. This condition will be satisfied by the choice of rows i_k in Sect. 5.2. Denote by

$$\|\mathfrak{J}_{k^*}\| := \max\{\|\mathfrak{J}_{k^*} f\|_\infty / \|f\|_\infty : f \in C(Y_t)\}$$

the Lebesgue constant of the interpolation operator \mathfrak{J}_{k^*} defined by

$$\mathfrak{J}_{k^*} f := \sum_{\ell=1}^{k^*} f(y_{j_\ell}) L_\ell$$

with

$$L_\ell(x) := \frac{\det W_\ell(x)}{\det W}, \quad \ell = 1, \dots, k^*,$$

being the Lagrange functions for Ξ_ℓ and y_ℓ , $\ell = 1, \dots, k^*$. Here, $W_\ell(x)$ denotes the matrix which results from replacing the ℓ -th row of W by the vector $[\Xi_j(x)]_{j=1, \dots, k^*}$. Since

$$f - \mathfrak{J}_{k^*} f = f - v + \mathfrak{J}_{k^*}(v - f) \quad \text{for all } v \in \text{span } \Xi,$$

it follows that, up to constants, the interpolation error $E_k(f) := f - \mathfrak{J}_{k^*} f$ is bounded by the error of the best approximation

$$\|E_k(f)\|_{\infty, Y_t} \leq (1 + \|\mathfrak{J}_{k^*}\|) \inf_{v \in \text{span } \Xi} \|f - v\|_{\infty, Y_t}. \quad (5.8)$$

Theorem 5.4. *Let $\mathcal{P}_i f = f(y_i)$, $i = 1, \dots, m$. Then for $i = 1, \dots, m$ and $j = 1, \dots, n$ it holds that*

$$|(R_k)_{ij}| \leq (1 + \|\mathfrak{J}_{k^*}\|)(1 + 2^k) E_{ts}^\Xi, \quad (5.9)$$

where E_{ts}^Ξ is defined in (5.6).

Proof. For the entries of the $(m - k) \times (n - k)$ matrix $A_{22} - X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix}$ it holds that

$$(A_{22} - X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix})_{ij} = \mathcal{K}Q_j^*(y_i) - \sum_{\ell=1}^{k^*} L_\ell(y_i) \mathcal{K}Q_j^*(y_\ell),$$

where we have chosen $X_{i\ell} = L_\ell(y_i)$. From (5.8) we have

$$|\mathcal{K}Q_j^*(y_i) - \sum_{\ell=1}^{k^*} L_\ell(y_i) \mathcal{K}Q_j^*(y_\ell)| \leq (1 + \|\mathfrak{J}_{k^*}\|) \inf_{v \in \text{span } \Xi} \|\mathcal{K}Q_j^* - v\|_{\infty, Y_t}.$$

The same kind of estimate holds for the entries of $A_{21} - X \begin{bmatrix} A_{11} \\ \hat{A}_{21} \end{bmatrix}$. Therefore, from Lemma 5.3 we obtain

$$|(R_k)_{ij}| \leq (1 + \|\mathfrak{J}_{k^*}\|)(1 + 2^k) \max_{j=1, \dots, n} \inf_{v \in \text{span } \Xi} \|\mathcal{K}Q_j^* - v\|_{\infty, Y_t}.$$

□

The previous theorem shows that the approximation computed by ACA is (up to constants) as good as the best approximation of the kernel function in any system of functions.

Galerkin matrices need a different treatment. For the linear operators \mathcal{P}_i , $i = 1, \dots, k^*$, corresponding to the first k^* rows in A we assume the following generalization of (5.7)

$$\det[\mathcal{P}_i \Xi_j]_{i,j=1,\dots,k^*} \neq 0, \quad (5.10)$$

which again will be guaranteed by the choice of pivoting rows i_k in Sect. 5.2. The operators \mathcal{P}_i in the case of Galerkin matrices read

$$\mathcal{P}_i f = \int_{\Omega} f(y) \psi_i(y) \, ds_y, \quad i = 1, \dots, m.$$

In the following lemma functions $\sum_{\ell=1}^{k^*} c_{\ell}^{(i)} \psi_{\ell} / \|\psi_{\ell}\|_{L^1}$ will be constructed such that some kind of *vanishing moments property*

$$\int_{\Omega} \left(\frac{\psi_i}{\|\psi_i\|_{L^1}} - \sum_{\ell=1}^{k^*} c_{\ell}^{(i)} \frac{\psi_{\ell}}{\|\psi_{\ell}\|_{L^1}} \right) v \, ds_x = 0 \quad \text{for all } v \in \text{span } \Xi \quad (5.11)$$

holds.

Lemma 5.5. *Assume that condition (5.10) is satisfied. Then for each $i \in \{1, \dots, m\}$ there are uniquely determined coefficients $c_{\ell}^{(i)}$, $\ell = 1, \dots, k^*$, such that (5.11) holds.*

Proof. For each $v \in \text{span } \Xi$ there are uniquely defined coefficients α_j , $j = 1, \dots, k^*$, such that

$$v = \sum_{j=1}^{k^*} \alpha_j \Xi_j. \quad (5.12)$$

Since the matrix $(\mathcal{P}_i \Xi_j)_{ij}$ is nonsingular, the linear system

$$\sum_{\ell=1}^{k^*} c_{\ell}^{(i)} \frac{\mathcal{P}_{\ell} \Xi_j}{\|\psi_{\ell}\|_{L^1}} = \frac{\mathcal{P}_i \Xi_j}{\|\psi_i\|_{L^1}}, \quad j = 1, \dots, k^*,$$

is uniquely solvable with respect to $c_{\ell}^{(i)}$, $\ell = 1, \dots, k^*$, for each $i \in \{1, \dots, m\}$. The assertion follows from (5.12) and the linearity of the operators \mathcal{P}_i . \square

The coefficients $c_{\ell}^{(i)}$ from the previous lemma depend on the shape of the grid, but they do depend neither on the kernel function κ nor on the size of the finite elements. In the following theorem the case of Galerkin matrices is treated.

Theorem 5.6. *Let $\mathcal{P}_i f = \int_{\Omega} f(y) \psi_i(y) \, ds_y$, $i = 1, \dots, m$. Then for $i = 1, \dots, m$ and $j = 1, \dots, n$ it holds that*

$$|(R_k)_{ij}| \leq (1 + \|\mathfrak{J}_{k^*}\|)(1 + 2^k) \left(1 + \sum_{\ell=1}^{k^*} |c_{\ell}^{(i)}| \right) \|\psi_i\|_{L^1} E_{ts}^{\Xi}. \quad (5.13)$$

Proof. Since $\mathfrak{J}_{k^*} \mathcal{K} \mathcal{Q}_j^* \in \text{span } \Xi$, according to Lemma 5.5 there are coefficients $c_{\ell}^{(i)}$, $\ell = 1, \dots, k^*$, such that

$$\int_{\Omega} \mathfrak{J}_{k^*} \mathcal{K} \mathcal{Q}_j^*(y) \frac{\psi_i(y)}{\|\psi_i\|_{L^1}} \, ds_y = \sum_{\ell=1}^{k^*} c_{\ell}^{(i)} \int_{\Omega} \mathfrak{J}_{k^*} \mathcal{K} \mathcal{Q}_j^*(y) \frac{\psi_{\ell}(y)}{\|\psi_{\ell}\|_{L^1}} \, ds_y.$$

For $X \in \mathbb{R}^{(m-k) \times k^*}$ we choose the matrix with entries

$$x_{i\ell} = \frac{\|\psi_i\|_{L^1}}{\|\psi_\ell\|_{L^1}} c_\ell^{(i)}, \quad i = 1, \dots, m-k, \ell = 1, \dots, k^*.$$

Then for the entries of the $(m-k) \times (n-k)$ matrix $A_{22} - X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix}$ it holds that

$$\begin{aligned} (A_{22} - X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix})_{ij} &= \int_{\Omega} \mathcal{K} \mathcal{Q}_j^*(y) \psi_i(y) \, ds_y - \sum_{\ell=1}^{k^*} c_\ell^{(i)} \frac{\|\psi_i\|_{L^1}}{\|\psi_\ell\|_{L^1}} \int_{\Omega} \mathcal{K} \mathcal{Q}_j^*(y) \psi_\ell(y) \, ds_y \\ &= \int_{\Omega} (\mathcal{I} - \mathfrak{J}_{k^*}) \mathcal{K} \mathcal{Q}_j^*(y) \psi_i(y) \, ds_y - \sum_{\ell=1}^{k^*} c_\ell^{(i)} \frac{\|\psi_i\|_{L^1}}{\|\psi_\ell\|_{L^1}} \int_{\Omega} (\mathcal{I} - \mathfrak{J}_{k^*}) \mathcal{K} \mathcal{Q}_j^*(y) \psi_\ell(y) \, ds_y \\ &= \int_{\Omega} E_{k^*} [\mathcal{K} \mathcal{Q}_j^*](y) \psi_i(y) \, ds_y - \sum_{\ell=1}^{k^*} c_\ell^{(i)} \frac{\|\psi_i\|_{L^1}}{\|\psi_\ell\|_{L^1}} \int_{\Omega} E_{k^*} [\mathcal{K} \mathcal{Q}_j^*](y) \psi_\ell(y) \, ds_y. \end{aligned}$$

From (5.8) we have

$$\int_{\Gamma} |E_{k^*} [\mathcal{K} \mathcal{Q}_j^*](y)| |\psi_i(y)| \, ds_y \leq (1 + \|\mathfrak{J}_{k^*}\|) \|\psi_i\|_{L^1} \inf_{v \in \text{span } \Xi} \|\mathcal{K} \mathcal{Q}_j^* - v\|_{\infty, Y_t}.$$

The same kind of estimate holds for $A_{21} - X \begin{bmatrix} A_{11} \\ \hat{A}_{21} \end{bmatrix}$. Therefore, from Lemma 5.3 we obtain

$$|(R_k)_{ij}| \leq (1 + \|\mathfrak{J}_{k^*}\|)(1 + 2^k) \|\psi_i\|_{L^1} \left(1 + \sum_{\ell=1}^{k^*} |c_\ell^{(i)}| \right) \max_{j=1, \dots, n} \inf_{v \in \text{span } \Xi} \|\mathcal{K} \mathcal{Q}_j^* - v\|_{\infty, Y_t}.$$

□

We have already pointed out the relation of ACA and the LU decomposition in [2]. It is known that during the pivoted LU decomposition the so-called growth of entries may happen. Note that this is reflected by the factor 2^k in (5.13). The growth of entries is however rarely observable in practice.

5.2 The right choice of rows

We have seen that the choice of columns j_k from condition (5.2) is important for the boundedness of the coefficient $A_{11}^{-1} A_{12}$ in the error estimate. The choice of the rows i_k will guarantee condition (5.7) or more generally (5.10), which the interpolation in the system Ξ relies on. Satisfying this condition is indispensable as can be seen from the following example.

Example 5.7. *The evaluation of the double-layer potential*

$$\int_{\Gamma} \frac{(n_x, x-y)}{|x-y|^3} \varphi_j(x) \, ds_x$$

at y_i , where φ_j are defined on the left cluster and y_i are located on the right cluster of Fig. 1, leads to a reducible matrix A having the structure

$$A = \begin{bmatrix} 0 & A_{12} \\ A_{21} & 0 \end{bmatrix}.$$

The zero block in the first block row of A is caused by the interaction of domain D_1 with D_3 and the other block by the interaction of domain D_2 with D_4 lying on a common plane, respectively. If ACA is applied to A with a starting pivot from the rows and columns of A_{12} , then the next nonzero pivot can only be found within A_{12} . Hence, the algorithm stays within A_{12} and the stopping criterion is not able to notice the untouched entries of A_{21} .

The reason for this is that the interpolation points y_{i_k} are exclusively chosen from the plane described by D_3 if ACA is applied to A_{12} . As a consequence, the Vandermonde determinant will vanish for this choice of interpolation points and the error estimate (5.9) does not hold for points $y_i \in D_4$. It remains however valid for points $y_i \in D_3$ since the interpolation problem reduces to a problem in \mathbb{R}^2 for which (5.7) can be guaranteed by an appropriate choice of $y_{i_k} \in D_3$.

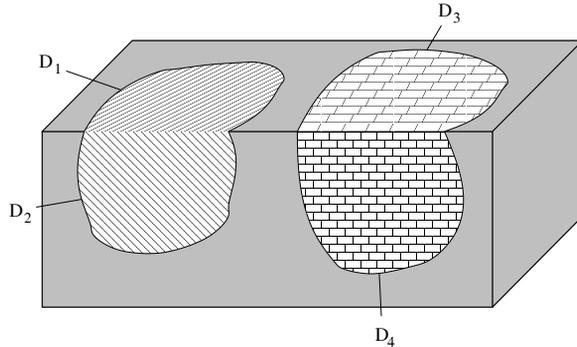


Figure 1: Two clusters leading to a reducible matrix block

The example above underlines the importance of the row choice, namely condition (5.10) in the convergence analysis of ACA. Assuming that ACA works without this condition may lead to a certain confusion; see [6].

We present two methods to choose the right rows. One of them should be used to circumvent the mentioned difficulty. The first is an easy and obvious heuristic which seems to work reliably. The second method explicitly guarantees (5.10) such that ACA will converge as predicted by our analysis.

For each row and each column of A we introduce a counter which reflects the number of successful approximations applied to the respective row or column. An approximation is considered as successful for row i if $|(u_k)_i|$ is of the order of the estimated error $\|R_{k-1}\|_F$. The counter of the column j is increased if $|(v_k)_j|$ has the size of $\|R_{k-1}\|_F$. With these counters one can easily detect those rows and columns for which the error estimator is not reliable. Pivoting to the latter rows and columns approximates the unattended rows and columns.

Although the problem described in Example 5.7 can be overcome by the mentioned trick, for a rigorous analysis that works in a general setting, we have to guarantee that (5.10) is satisfied by the choice of pivots. The first row i_1 can be arbitrarily chosen, because we have assumed that $\Xi_1(x) = 1$. Assume that $i_1, \dots, i_k \in \{1, \dots, m\}$ and $j'_1, \dots, j'_k \in \{1, \dots, n\}$ have already been found such that (5.10) holds; i.e., the Vandermonde matrix $W_k \in \mathbb{R}^{k \times k}$ having the entries

$$(W_k)_{\mu\nu} = \mathcal{P}_{i_\mu} \Xi_{j'_\nu}, \quad \mu, \nu = 1, \dots, k,$$

is nonsingular. Note that the indices j'_k may differ from the indices j_k used in the previous section. The new pivot (i_{k+1}, j'_{k+1}) has to guarantee that W_{k+1} is nonsingular, too.

Assume that the normalized LU decomposition of the the $k \times k$ matrix $W_k = L_k U_k$ has been computed. Then

$$\det W_{k+1} = \left[c - (U_k^{-T} a)^T (L_k^{-1} b) \right] \det W_k,$$

where

$$W_{k+1} = \begin{bmatrix} W_k & b \\ a^T & c \end{bmatrix}$$

and $a = [\mathcal{P}_{i_{k+1}} \Xi_{j'_\ell}]_{\ell=1, \dots, k}$, $b = [\mathcal{P}_{i_\ell} \Xi_{j'_{k+1}}]_{\ell=1, \dots, k}$, and $c = \mathcal{P}_{i_{k+1}} \Xi_{j'_{k+1}}$. Hence, any pair (i_{k+1}, j'_{k+1}) satisfying $(U_k^{-T} a)^T (L_k^{-1} b) \neq c$ can be chosen as the new pivot. For stability it is however wise to choose (i_{k+1}, j'_{k+1}) such that $|c - (U_k^{-T} a)^T (L_k^{-1} b)|$ is maximized. Testing a new row i_{k+1} requires k^2 operations. Since possibly all remaining $m - k$ rows have to be tested, finding i_{k+1} requires $k^2(m - k)$ operations. The LU decomposition of W_{k+1} is then given by

$$W_{k+1} = \begin{bmatrix} L_k & 0 \\ a^T U_k^{-1} & 1 \end{bmatrix} \begin{bmatrix} U_k & L_k^{-1} b \\ 0 & c - (U_k^{-T} a)^T (L_k^{-1} b) \end{bmatrix}.$$

Example 5.8. We return to the problem from Example 5.7. Assume that three pivots have been chosen from D_3 . If i_4 is chosen from the same set, then one easily calculates that

$$\det[\Xi_j(y_{i_\mu})]_{1 \leq \mu, j \leq 4} = 0,$$

where $\Xi_1(x) = 1$, $\Xi_2(x) = x_1$, $\Xi_3(x) = x_2$, and $\Xi_4(x) = x_3$. There are however points $y_{i_4} \in D_4$ which will satisfy $\det[\Xi_j(y_{i_\mu})]_{1 \leq \mu, j \leq 4} \neq 0$. Hence, ACA will automatically choose the fourth pivot from the row indices of the block A_{21} .

It may happen that columns of $\mathcal{P}_i \Xi_j$ are linearly dependent. If $\mathcal{P}_i f = f(y_i)$ with points y_i , $i = 1, \dots, m$, then this is equivalent to all points y_i lying on a hypersurface $H := \{x \in \mathbb{R}^d : v_H(x) = 0\}$, where $v_H \in \text{span } \Xi$. In such a situation, for certain j no $i_{k+1} \in \{1, \dots, m\}$ can be found such that $(U_k^{-T} a)^T (L_k^{-1} b) \neq c$. Let j'_{k+1} be the first column which is not in the linear hull of the first k . Then search for i_{k+1} in the $m \times (k + 1)$ matrix which contains the j'_{k+1} -st column of A as its last column. Note that the number of investigated columns is bounded by the number of columns required in the case of linear independent columns. This can be seen from

$$\text{span } \Xi = \text{span } \Xi' \quad \text{on } H,$$

where $\Xi' := \{\Xi_{j'_1}, \dots, \Xi_{j'_{k'}}\}$. The last equation shows that k' columns give the same error E_{ts}^{Ξ} as all k^* . The previous arguments also hold if $\mathcal{P}_i f = \int_{\Omega} f \psi_i ds$.

As a result of the above procedure, we obtain two sequences i_ℓ and j'_ℓ , $\ell = 1, \dots, k'$, such that (5.10) is satisfied; i.e.,

$$\det[\mathcal{P}_{i_\mu} \Xi_{j'_\nu}]_{1 \leq \mu, \nu \leq k'} \neq 0.$$

Example 5.9. Let $d = 2$ and $\Xi = \{1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, \dots\}$. If all points y_i are located on the x -axis, then $j'_1 = 1$, $j'_2 = 2$, $j'_3 = 4$ and so on. Hence, ACA will detect the reduced dimensionality of the problem and will implicitly use $\Xi' := \{1, x_1, x_1^2, \dots\}$.

Note that the above pivoting criterion requires only geometrical information and information about \mathcal{P}_i , i.e, whether a collocation or a Galerkin method is employed. The kernel function, however, is still not required. The method proposed in [6] is based on explicit kernel approximation such that the original matrix entries cannot be used.

We have seen that the first row i_1 can be arbitrarily chosen. The efficiency of ACA can however be improved by the following choice. Due to the Definition 3.1 of asymptotic smoothness, each kernel function $\kappa(x, y)$ is almost constant with respect to y on $X_s \times Y_t$. Hence, if the expression

$$\max_{y \in Y_t} |\kappa(x, y) - \kappa(x, z)| = \max_{y \in Y_t} \left| \int_y^z \partial_y \kappa(x, \xi) d\xi \right| \leq c \max_{y \in Y_t} |y - z|$$

appearing in the remainder after the first step of ACA is to be minimized with respect to z , one should choose z to be the Chebyshev center of Y_t . Since the Chebyshev center of a polygonal set is quite expensive to compute, we use the centroid m_t of Y_t instead. Hence, from these arguments it seems promising to choose i_1 so that the center z_{i_1} of the i_1 -st support Y_{i_1} is closest to m_t .

In Figure 2 we compare this strategy with the “old” strategy in which i_1 is chosen so that z_{i_1} is closest to the centroid of X_s . The matrix which these methods are applied to arises from evaluating $|x - y|^{-1}$ at two sets of points having a distance which is large compared with their diameters. In addition, Figure 2 shows the accuracy of approximations obtained from the multipole expansion and from the singular value decomposition applied to an admissible matrix block. The first k

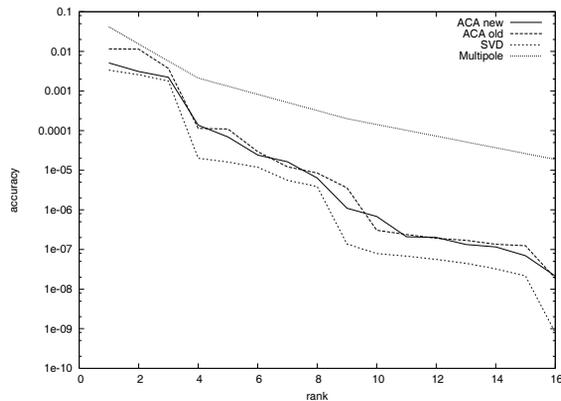


Figure 2: Accuracy of different approximants.

singular triplets give provably the best accuracy among all rank- k approximants. The influence of the new strategy on the quality of the approximation can be realized in the first four steps, which are often sufficient to obtain a reasonable accuracy. In this part the new version gives almost optimal approximation. For all other steps the old and the new version behave almost the same. The quality of the approximant generated from the multipole expansion is significantly worse.

5.3 Overall complexity

Finally, we will estimate the overall cost of generating the approximant and multiplying it by a vector if an accuracy $\varepsilon > 0$ for the approximation error

$$\|K - K_{\mathcal{H}}\|_F < \varepsilon \|K\|_F$$

is prescribed. Due to the properties of the Frobenius norm, the last condition is implied by the condition

$$\|K_b - S_k\|_F < \varepsilon \|K_b\|_F \quad (5.14)$$

on each block $b \in P$. Since the error estimates (5.9) and (5.13) depend on the choice of the approximation system Ξ , we cannot estimate the constants appearing in these estimates. Since they will enter the overall complexity through the logarithm anyhow, we assume that they are bounded independently of m , n , and k . According to (5.6), an asymptotically smooth kernel κ on a pair of domains satisfying (2.1) can be approximated by polynomials of order p with accuracy η^p . Since boundary element methods are applied to $(d - 1)$ -dimensional manifolds in \mathbb{R}^d and since ACA can detect the reduced dimensionality, we will assume that for each admissible block $K_b \in \mathbb{R}^{t \times s}$ the approximant S_k , $k = \dim \Pi_p^{d-1} \sim p^{d-1}$, satisfies

$$\|K_b - S_k\|_F \leq c \eta^p \|K_b\|_F.$$

If a block does not satisfy (2.1), then its original entries are stored. For

$$p \geq \log_{1/\eta} c/\varepsilon$$

we obtain condition (5.14).

The complexity of an \mathcal{H} -matrix results from the blockwise costs in a way that was described in Sect. 2. Since $k \sim p^{d-1}$, generating an approximant $\tilde{A} \in \mathcal{H}(P, k)$ requires $N \log N |\log \varepsilon|^{2(d-1)}$ arithmetical operations. Once $\tilde{A}_{\mathcal{H}}$ has been generated, storing and multiplying it by a vector can be done with complexity $N \log N |\log \varepsilon|^{d-1}$.

6 Numerical experiments

In this section we are going to test Algorithm 5.1 on the boundary integral formulation of the inner mixed boundary value problem for the Lamé system (4.1) in three dimensions.

We reorder the unknown coefficients to separate the components

$$\begin{aligned} \{u_{x,i}, u_{y,i}, u_{z,i}\}_{i=1}^N &\rightarrow \left\{ \{u_{x,i}\}_{i=1}^N, \{u_{y,i}\}_{i=1}^N, \{u_{z,i}\}_{i=1}^N \right\}, \\ \{t_{x,j}, t_{y,j}, t_{z,j}\}_{j=1}^{N'} &\rightarrow \left\{ \{t_{x,j}\}_{j=1}^{N'}, \{t_{y,j}\}_{j=1}^{N'}, \{t_{z,j}\}_{j=1}^{N'} \right\}. \end{aligned}$$

In this setting, the matrix V_h can be decomposed as

$$V_h = c_1 \begin{bmatrix} (V_L)_h & 0 & 0 \\ 0 & (V_L)_h & 0 \\ 0 & 0 & (V_L)_h \end{bmatrix} + c_2 \begin{bmatrix} V_1 & V_2 & V_3 \\ V_2 & V_4 & V_5 \\ V_3 & V_5 & V_6 \end{bmatrix},$$

where $(V_L)_h$ is the discretization of the single-layer potential for the Laplace equation and V_i , $i = 1, \dots, 6$, are the matrices that correspond to the entries of the second term in (4.5).

Thus, for the solution of the system (4.8) we need to store seven Hermitian $N \times N$ matrices to represent V_h and one $N \times N'$ matrix K_L . These matrices are composed using the hierarchical clustering and the Algorithm 5.1 to approximate admissible blocks. We first compute the matrices for the geometry depicted in Fig. 3. Nodes on the left face of the beam are fixed ($u = 0$) and a unit vertical force $t = (0, 0, -1)$ is applied to each triangle of the right face. The mesh is uniformly refined to test the performance on matrices of different sizes. The compression results for the matrix

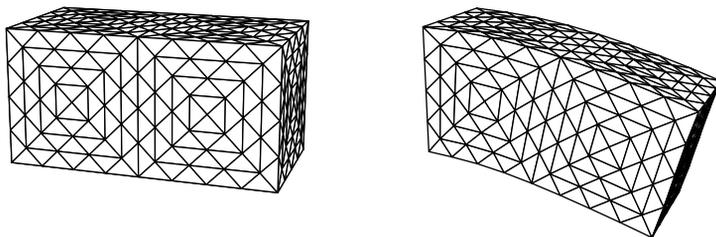


Figure 3: Test domain before and after the elastic deformation.

V_h are shown in Table 1 for fixed $\varepsilon = 10^{-5}$. As expected, the compression ratio (column 3) decays. Furthermore, we observe a decay of the memory increment (column 4). This reflects the almost linear storage requirement for the matrix. The amount of computed matrix entries is shown in column 5 of the table. Column 6 shows that the compression procedure uses only a small portion of matrix entries to build a blockwise low-rank approximant.

The required CPU time per iteration also behaves in an adequate way; see Table 2. The decay of the time increment (column 3) indicates the almost linear complexity of the compressed matrix-vector multiplication procedure.

N	size [MB]	ratio	mem. inc.	entr. [mill.]	gain	entr. inc.
640	9.50	0.338	–	2.7	0.738	–
1 280	30.87	0.274	3.250	8.8	0.600	3.250
2 560	90.37	0.201	2.928	25.8	0.439	2.930
5 120	239.49	0.133	2.650	64.8	0.275	2.510
10 240	562.58	0.078	2.349	141.6	0.150	2.182
20 480	1 291.78	0.045	2.296	358.5	0.099	2.531
40 960	2 624.02	0.023	2.031	820.4	0.054	2.288

Table 1: Compression and number of computed elements of the matrix V_h for $\varepsilon = 10^{-5}$.

N	iter. time [sec]	iter. time inc.
640	0.034	–
1 280	0.128	3.802
2 560	0.357	2.787
5 120	0.930	2.604
10 240	2.229	2.395
20 480	4.905	2.201
40 960	10.659	2.173

Table 2: The time needed for one iteration when inverting compressed V_h for $\varepsilon = 10^{-5}$.

Table 3 shows how the compression rate of the matrix V_h depends on the approximation precision ε . Here $N = 20480$, and matrix dimensions are $3N \times 3N$.

ε	size [MB]	ratio
1E-2	458.01	0.0159
1E-3	633.51	0.0220
1E-4	848.91	0.0295
1E-5	1 291.78	0.0449

Table 3: The size of compressed matrix V_h depending on approximation precision ε ($N = 20480$).

The next test is performed on the geometry shown in Fig. 4. Nodes on the bottom face of the body are fixed ($u = 0$) and a unit vertical force $t = (0, 0, 1)$ is applied to each triangle of the top face. The performance results are collected in Table 4. We observe a behavior similar to the behavior in the previous test.

References

- [1] S. Agmon: *Lectures on elliptic boundary value problems*. N. J. Van Nostrand, Princeton 1965.
- [2] M. Bebendorf: *Approximation of boundary element matrices*. Numer. Math. 86, 565–589, 2000.

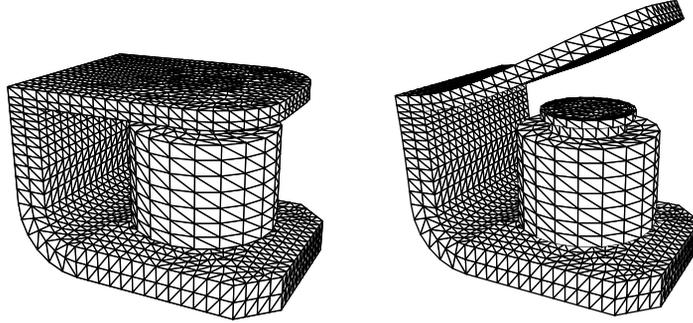


Figure 4: Test domain before and after the elastic deformation.

N	size [MB]	ratio	mem. inc.	entr. [mill.]	gain	entr. inc.	cpu [sec]	time inc.
4 944	278.26	0.1658	–	61	0.275	–	2 441	–
19 776	1 791.83	0.0667	6.439	361	0.102	5.960	12 658	5.186
79 104	9 499.65	0.0221	5.302	1786	0.032	4.951	46 941	3.708

Table 4: Compression and number of computed elements of the matrix V_h for the second test geometry ($\varepsilon = 10^{-4}$).

- [3] M. Bebendorf and S. Rjasanow: *Adaptive Low-Rank Approximation of Collocation Matrices*. Computing 70(1), 1–24, 2003.
- [4] M. Bebendorf and R. Kriemann: *Fast Parallel Solution of Integral equations and related problems*. Comput. Visual. Sci. 8, 121–135, 2005.
- [5] M. Bebendorf: *Hierarchical LU Decomposition-based Preconditioners for BEM*, Computing 74, 225–247, 2005.
- [6] S. Börm and L. Grasedyck: *Hybrid Cross Approximation of Integral Operators*. Numerische Mathematik 101(2), 221–249, 2005.
- [7] A. Buchau, W. Rucker, O. Rain, V. Rischmüller, S. Kurz, and S. Rjasanow: *Comparison between different approaches for fast and efficient 3-D BEM computations*. IEEE Transactions on Magnetics 39(3), 1107–1110, 2003.
- [8] M. Giaquinta: *Multiple Integrals in the calculus of variations and nonlinear elliptic systems*. Princeton University press, 1983.
- [9] S. A. Goreinov, E. E. Tyrtysnikov, and N. L. Zamarashkin: A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.
- [10] L. Grasedyck and W. Hackbusch, *Construction and arithmetics of \mathcal{H} -matrices*, Computing 70 (2003), pp. 295–334.
- [11] L. Greengard and V. Rokhlin: *A new version of the fast multipole method for the Laplace equation in three dimensions*. Acta numerica, 1997, pages 229–269. Cambridge Univ. Press, Cambridge, 1997.

- [12] W. Hackbusch: *A sparse matrix arithmetic based on \mathcal{H} -matrices. I. Introduction to \mathcal{H} -matrices.* Computing, 62(2):89–108, 1999.
- [13] W. Hackbusch and B. N. Khoromskij. *A sparse \mathcal{H} -matrix arithmetic. II. Application to multi-dimensional problems.* Computing, 64(1):21–47, 2000.
- [14] W. Hackbusch and Z. P. Nowak. *On the fast matrix multiplication in the boundary element method by panel clustering.* Numer. Math., 54(4):463–491, 1989.
- [15] K. Hayami and S. A. Sauter, *A panel clustering method for 3-D elastostatics using spherical harmonics.* Integral methods in science and engineering (Houghton, MI, 1998), 179–184, Chapman & Hall/CRC Res. Notes Math., 418, Chapman & Hall/CRC, Boca Raton, FL, 2000.
- [16] R. A. Horn and C. R. Johnson: *Matrix Analysis.* Corrected reprint of the 1985 original, Cambridge Univ. Press, Cambridge, 1990.
- [17] V. Kupradze: *Three-dimensional problems of the mathematical theory of elasticity and thermoelasticity.* North-Holland, Amsterdam, 1979.
- [18] S. Kurz, O. Rain, and S. Rjasanow: *The Adaptive Cross Approximation Technique for the 3D Boundary Element Method.* IEEE Transactions on Magnetics 38(2), 421–424, 2002.
- [19] U. Langer and D. Pusch: *Data-sparse algebraic multigrid methods for large scale boundary element equations.* Appl. Numer. Math. 54 (2005), no. 3-4, 406–424.
- [20] W. McLean: *Strongly Elliptic Systems and Boundary Integral Equations,* Cambridge University Press, 2000.
- [21] G. Of, O. Steinbach, and W. L. Wendland: *Applications of a fast multipole Galerkin boundary element method in linear elastostatics.* Comput. Vis. Sci. 8, 201–209, 2005.
- [22] V. Rokhlin: *Rapid solution of integral equations of classical potential theory.* J. Comput. Phys., 60(2):187–207, 1985.
- [23] T. Sauer and Y. Xu: *On multivariate Lagrange interpolation.* Math. Comp. 64(211):1147–1170, 1995.
- [24] S. A. Sauter and Ch. Schwab: *Randelementmethoden,* Teubner, 2004.
- [25] S. Sirtori, G. Maier, G. Novati and S. Miccoli: *A Galerkin symmetric boundary-element method in elasticity: formulation and implementation* Internat. J. Numer. Methods Engrg. 35(2), 255–282, 1992.
- [26] O. Steinbach: *Numerische Näherungsverfahren für elliptische Randwertprobleme: Finite Elemente und Randelemente* B. G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2003.
- [27] E. E. Tyrtshnikov: *Mosaic-skeleton approximations.* Calcolo, 33(1-2):47–57 (1998), 1996.
- [28] S. A. Goreinov, E. E. Tyrtshnikov, and N. L. Zamarashkin: *A theory of pseudoskeleton approximations.* Linear Algebra Appl. 261, 1–21, 1997.