Graz University of Technology
Institute of Applied Mechanics

Preprint Series

Preprint No 01/2018

Adaptive Cross Approximation for BEM in elasticity

Anita M. Haider, Martin Schanz
Institute of Applied Mechanics, Graz University of Technology

Published in *Journal of Theoretical and Computational Acoustics*, 27(01), 1850060, 2019
DOI: 10.1142/S2591728518500603
Latest revision: May 14, 2019

Abstract

Boundary element methods (BEM) have obtained a mature state in the last years so that industrial applications are possible. However, to treat real world problems so-called fast methods are necessary to reduce the original quadratic complexity to an almost linear order. Essentially, two methods are popular, the so-called fast multipole method, which uses a kernel expansion, and the algebraical approach based on $\mathcal{H}$-matrices with the adaptive cross approximation (ACA) to compress the matrix blocks. The latter is frequently used for scalar-valued problems, but for vector-valued problems a modification of the pivot strategy is required. It has been suggested to search for the largest singular value out of all minimal singular values of the fundamental solution blocks. This strategy has been proposed by Rjasanow and Weggler and is studied here for elastostatics and elastodynamics. It is shown with numerical experiments that this strategy is mostly robust and results in an almost linear complexity.

1 Introduction

The boundary element method (BEM) is well developed for many problem classes in engineering. In the present work, this method is applied on the numerical solution of boundary value problems in 3-d elastostatics and elastodynamics. The textbook such as Ref. [11] provides a comprehensive review of BEM applications in elasticity. The first boundary integral formulation for elastodynamics in Laplace or Fourier domain has been published by Cruse and Rizzo [15] and Domínguez [16]. A detailed review on elastodynamic boundary element formulations can be found in the articles of Beskos [7, 8] and Costabel [14]. For both, elastostatics and elastodynamics collocation based BE formulations are available (e.g. those referenced above) and Galerkin based formulations as well, e.g. Ref. [9].

With the improved capabilities of computer systems, larger problems can be solved. However, by increasing the size of the problem, the effort of solving fully populated matrices scales quadratically. Even though better computer hardware exists, the BEM reaches its limits. Hence, fast methods have become popular in the field of applied mathematics and engineering in particular. The origins of such methods, i.e. asymptotically optimal approximations of fully populated matrices, can be traced back to Rokhlin [38]. For the first time an algorithm was presented which scales like $O(N \log N)$, where N is the number of degrees of freedom. Subsequently, the fast multipole method (FMM) has been developed in Ref. [22] for some large-scale N-body problems and has been significantly improved in Ref. [23]. In the work of Of et al. [35] the FMM has been applied to elastostatic problems based on a Galerkin BEM discretization. The extension to elastodynamics in Fourier domain has been published in Ref. [12] based on a collocation approach. A black box FMM approach for scalar-valued problems has been proposed by Fong and Darve [18]. Other approaches are panel clustering (see Ref. [27]) and the wavelet based BEM [1]. The latter method produces sparse matrices based on orthogonal systems of wavelet like functions.

All these methodologies perform matrix-vector multiplications in almost linear complexity. However, the only approach that enables all matrix operations (matrix-vector product, matrix-matrix product, matrix-matrix addition, matrix inversion, LU decomposition, etc.) of almost linear complexity are the so-called $\mathcal{H}$-matrices introduced by Hackbusch [25]. They can be understood as algebraic structure reflecting a geometrically motivated partitioning into subblocks. Each subblock is classified to be either admissible or not. A further development of the $\mathcal{H}$-matrices are the $\mathcal{H}^2$-matrices [26].

After constructing an $\mathcal{H}$-matrix, admissible blocks are approximated. All previously mentioned methods, such as FMM, panel clustering, and wavelet based methods approximate discrete integral operators in a very specific way. They deal with the analytical decomposition of integral kernels and, hence, the procedure becomes problem dependent. This fact holds also for the coding of this class of methodologies. A second class are the so-called algebraic approximation methods. The singular value decomposition (SVD) leads to the optimal low-rank approximation, however, with $O(N^3)$ complexity. Less expensive algorithms are the mosaic skeleton method developed in Ref. [20] and the successively developed adaptive cross approximation (ACA) [3]. In the present work, the application of the ACA to elastic problems is examined. It has been applied by Bebendorf and Rjasanow [2, 5] to the approximation of BEM matrices for the first time. The outstanding feature of ACA compared to SVD is that it requires only the

evaluation of some original matrix entries and the approximation is still almost optimal. For this reason, it can be used in a black box like manner. Its coding and adaptation to existing codes is straightforward. The algorithm is robust and it is based on a stopping criterion depending on a prescribed approximation accuracy ε. In elasticity, the ACA has been successfully applied to the solution of mixed elastostatic boundary value problems by Bebendorf and Grzhibovskis [4]. In this work, an error estimate for approximated Galerkin matrices has been presented. Furthermore, an improved pivoting strategy is given, such that the ACA algorithm will not fail in some special cases. To overcome such problems a variant called hybrid cross approximation (HCA) has been developed by Börm and Grasedyck [10]. An engineering approach for the acceleration of elastostatic problems has been presented in Ref. [31]. In Refs. [6] and [24], the approximation of $\mathcal{H}$-matrices generated by the ACA has been efficiently applied to crack problems in elastic media solved by using a collocation boundary element formulation.

This listing suggests that ACA based BE formulations are readily available for applications. However, a crucial point remains to be discussed for vector-valued problems. To use ACA efficiently with respect to CPU time, a partially-pivoted algorithm must be used. The choice of the pivot element is not obvious for vector-valued problems like elasticity. The ACA algorithm assumes that the approximated function is smooth, which is not true if the matrix entries for the different directions are considered. For each direction it would hold and, therefore, in Ref. [34] the overall matrix has been partitioned in each coordinate direction. This pivot strategy is suboptimal. An improved strategy is to find a suitable pivot element consisting of all entries of the fundamental solution. This has been proposed and discussed by Rjasanow and Weggler [37] for electromagnetic problems and later used in Ref. [13] for elastodynamics.

This work is dedicated towards applying the improved pivot strategy of the partially-pivoted ACA to elastostatics and elastodynamics. As well, the studies concerning the accuracy in relation to efficiency are shown. In the following, a collocation approach is used as this is the most common approach in engineering.

2 Governing Equations

The principal objective in this section is to formulate the basic equations of three-dimensional elastic waves, its static specialisation, and the corresponding boundary integral and algebraic equations. To obtain the elastodynamic equation Cauchy's first law of motion and the theory of linear elasticity is used.

2.1 Problem setting

Consider a continuous homogeneous isotropic elastic body $\Omega \subset \mathbb{R}^3$ with a Lipschitz boundary $\Gamma := \partial\Omega$. Throughout this paper, we treat the elastodynamic wave propagation in the Laplace-domain in the absence of body forces. Assuming vanishing initial condition, the elastodynamic wave equation for the unknown displacement field $\mathbf{u}(\mathbf{x})$ is the Lamé-Navier equation

$$\mu\Delta\mathbf{u}(\mathbf{x},s) + (\lambda+\mu)\,\mathrm{grad}\,\mathrm{div}\,\mathbf{u}(\mathbf{x},s) - \rho s^2\mathbf{u}(\mathbf{x},s) = \mathbf{0} \quad \mathbf{x}\in\Omega \tag{1}$$

with the Laplace parameter $s \in \mathbb{C}$ satisfying $\Re(s) > 0$. The mass density of the body is denoted by ρ, and λ and μ are the Lamé constants that are related to the elasticity modulus E and the

Poisson ratio ν of the material

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

In engineering, the elastodynamic equation in frequency-domain is more frequently employed by using a Fourier transform instead of Laplace transform. We gain the displacement equation of motion for time-harmonic problems by replacing s in (1) with $i\omega$, where ω is the angular frequency.

On the boundary we take given displacement data $\mathbf{g}_D$ on Γ_D and traction data $\mathbf{g}_N$ on Γ_N, respectively, sucht that $\Gamma = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Thus, the boundary conditions are given by

$$\mathbf{u}(\mathbf{x}) = \mathbf{g}_D(\mathbf{x}) \quad \mathbf{x} \in \Gamma_D, \tag{3}$$

$$\mathbf{t}(\mathbf{x}) := \mathcal{T}_{\mathbf{x}}\mathbf{u}(\mathbf{x}) = \mathbf{g}_N(\mathbf{x}) \quad \mathbf{x} \in \Gamma_N. \tag{4}$$

The Neumann trace for elastodynamic problems is described by the traction operator

$$(\mathcal{T}_{\mathbf{x}}\mathbf{u})(\mathbf{x}) = \lambda \operatorname{div}\mathbf{u}(\mathbf{x})\mathbf{n}(\mathbf{x}) + 2\mu\frac{\partial}{\partial\mathbf{n}}\mathbf{u}(\mathbf{x}) + \mu\mathbf{n}(\mathbf{x}) \times \operatorname{curl}\mathbf{u}(\mathbf{x}). \tag{5}$$

If the inertia term in (1) vanishes, the equation reduces to the displacement equation of elastostatics

$$\mu\Delta\mathbf{u}(\mathbf{x}) + (\lambda+\mu)\operatorname{grad}\operatorname{div}\mathbf{u}(\mathbf{x}) = \mathbf{0} \quad \mathbf{x} \in \Omega. \tag{6}$$

2.2 Boundary integral formulations

The unknown displacements in the domain are related to the boundary data by the representation formula, also known as the Somigliana identity

$$\mathbf{u}(\widetilde{\mathbf{x}}) = \int_\Gamma \mathbf{U}(\widetilde{\mathbf{x}},\mathbf{y})\mathbf{t}(\mathbf{y})\mathrm{d}s_{\mathbf{y}} - \int_\Gamma (\mathcal{T}_{\mathbf{y}}\mathbf{U})(\widetilde{\mathbf{x}},\mathbf{y})\mathbf{u}(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \quad \widetilde{\mathbf{x}} \in \Omega\,. \tag{7}$$

The displacement and traction fundamental solution are denoted with $\mathbf{U}(\widetilde{\mathbf{x}},\mathbf{y})$ and $(\mathcal{T}_{\mathbf{y}}\mathbf{U})(\widetilde{\mathbf{x}},\mathbf{y})$, respectively, and can be found, e.g. in Refs. [15] or [19]. Applying the Dirichlet trace to the representation formula (7) and using the operator notation we obtain the first boundary integral equation

$$\mathcal{C}(\mathbf{x})\mathbf{u}(\mathbf{x}) = (\mathcal{V}\mathbf{t})(\mathbf{x}) - (\mathcal{K}\mathbf{u})(\mathbf{x}) \quad \mathbf{x} \in \Gamma. \tag{8}$$

The introduced operators are the single layer operator $\mathcal{V}$, the double layer operator $\mathcal{K}$ and the integral-free term $\mathcal{C}$. Aside from the direct boundary integral equation (8) where the integral operators acts directly on the boundary data, the so-called indirect approach exists. For this approach, the same integral operators are applied to more or less arbitrary density functions.

Given an arbitrary density function $\mathbf{w}$ and $\mathbf{v}$ we define the single layer potential (SLP)

$$\mathbf{u}(\widetilde{\mathbf{x}}) = \left(\widetilde{\mathcal{V}}\mathbf{w}\right)(\widetilde{\mathbf{x}}) = \int_\Gamma \mathbf{U}(\widetilde{\mathbf{x}},\mathbf{y})\mathbf{w}(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \quad \widetilde{\mathbf{x}} \in \Omega \tag{9}$$

and the double layer potential (DLP)

$$\mathbf{u}(\widetilde{\mathbf{x}}) = \left(\widetilde{\mathcal{K}}\mathbf{v}\right)(\widetilde{\mathbf{x}}) = \int_{\Gamma} \left(\mathcal{T}_{\mathbf{y}}\mathbf{U}\right)(\widetilde{\mathbf{x}},\mathbf{y})\mathbf{v}(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \quad \widetilde{\mathbf{x}} \in \Omega\,. \tag{10}$$

Taking the Dirichlet trace γ_0 of (9) for $\mathbf{x} \in \Gamma$ results in the weakly singular boundary integral operator

$$\left(\mathcal{V}\mathbf{w}\right)(\mathbf{x}) \coloneqq \gamma_0\left(\widetilde{\mathcal{V}}\mathbf{w}\right)(\mathbf{x}) = \int_{\Gamma} \mathbf{U}(\mathbf{x},\mathbf{y})\mathbf{w}(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \quad \mathbf{x} \in \Gamma. \tag{11}$$

By applying the Dirichlet trace operator to the double layer potential (10) we have the representation

$$\gamma_0\left(\widetilde{\mathcal{K}}\mathbf{v}\right)(\mathbf{x}) = \left(-\mathbf{I} + \mathcal{C}(\mathbf{x})\right)\mathbf{v}(\mathbf{x}) + \left(\mathcal{K}\mathbf{v}\right)(\mathbf{x}) \quad \mathbf{x} \in \Gamma \tag{12}$$

with the double layer operator

$$\left(\mathcal{K}\mathbf{v}\right)(\mathbf{x}) \coloneqq \lim_{\varepsilon \to 0} \int_{\mathbf{y}\in\Gamma : |\mathbf{y}-\mathbf{x}| \geq \varepsilon} \left(\mathcal{T}_{\mathbf{y}}\mathbf{U}\right)(\mathbf{x},\mathbf{y})\mathbf{v}(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \quad \mathbf{x} \in \Gamma. \tag{13}$$

The computation of the integral free term $\mathcal{C}$ is given in detail in the work of Mantič [32] or Hartmann [29]. The above sketched formulations are formally equivalent for elastodynamics and elastostatics but different fundamental solutions have to be used. Clearly, in the latter case there is no dependency on the Laplace parameter s.

2.3 Spatial discretization

The governing equations are discussed in Laplace domain, therefore only a spatial discretization has to be considered. To solve the boundary integral equation the boundary is decomposed in a finite number of triangles

$$\Gamma \approx \Gamma_h = \bigcup_{k=1}^{N_E} \overline{\tau_k}, \tag{14}$$

where N_E denotes the total number of elements. We approximate the traction field $\mathbf{t}$ and the density $\mathbf{w}$ by constant discontinuous functions φ^0

$$\mathbf{t}(\mathbf{x}) \approx \mathbf{t}_h(\mathbf{x}) = \sum_{j=1}^{N_e} \varphi_j^0(\mathbf{x})\mathbf{t}_j, \quad \mathbf{w}(\mathbf{x}) \approx \mathbf{w}_h(\mathbf{x}) = \sum_{j=1}^{N_e} \varphi_j^0(\mathbf{x})\mathbf{w}_j, \tag{15}$$

and the displacement field $\mathbf{u}$ and the density $\mathbf{v}$ by linear continuous functions φ^1

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}_h(\mathbf{x}) = \sum_{j=1}^{N_n} \varphi_j^1(\mathbf{x})\mathbf{u}_j, \quad \mathbf{v}(\mathbf{x}) \approx \mathbf{v}_h(\mathbf{x}) = \sum_{j=1}^{N_n} \varphi_j^1(\mathbf{x})\mathbf{v}_j. \tag{16}$$

In order to get a fully discrete representation of the boundary integral equation, we make use of the collocation method. For the unknown tractions we take the collocation point $\mathbf{x}_i$ at the center of the element and for the unknown displacements we collocate at the element nodes.

Therefore, N_e is the number of elements and N_n the number of vertices. Consequently, the discretized integral equation of the direct representation formula (8) is

$$\mathcal{C}(\mathbf{x}_i)\mathbf{u}(\mathbf{x}_i) = \sum_{j=1}^{N_e} \int\limits_{\mathrm{supp}(\varphi_j^0)} \mathbf{U}(\mathbf{x}_i,\mathbf{y})\varphi_j^0(\mathbf{y})\mathrm{d}s_{\mathbf{y}}\mathbf{t}_j - \sum_{j=1}^{N_n} \fint\limits_{\mathrm{supp}(\varphi_j^1)} (\mathcal{T}_{\mathbf{y}}\mathbf{U})(\mathbf{x}_i,\mathbf{y})\varphi_j^1(\mathbf{y})\mathrm{d}s_{\mathbf{y}}\mathbf{u}_j \,. \tag{17}$$

The integral over the traction fundamental solution exists only in the sense of a Cauchy principal value because of its strong singularity and is denoted by the symbol $\fint$. Note that the displacement and surface traction field are vector fields and, therefore, $\mathbf{u}_j, \mathbf{t}_j \in \mathbb{K}^3$. If we consider an elastostatic problem the entries are real-valued, i.e. $\mathbb{K} = \mathbb{R}$ and in the case of elastodynamics the field variables are complex-valued, meaning $\mathbb{K} = \mathbb{C}$.

Regarding the indirect approach the discrete version of the SLP is

$$\mathbf{u}(\mathbf{x}_i) = \sum_{j=1}^{N_e} \int\limits_{\mathrm{supp}(\varphi_j^0)} \mathbf{U}(\mathbf{x}_i,\mathbf{y})\varphi_j^0(\mathbf{y})\mathrm{d}s_{\mathbf{y}}\mathbf{w}_j \,, \tag{18}$$

and that of the DLP reads as

$$\mathbf{u}(\mathbf{x}_i) = \sum_{j=1}^{N_n} \fint\limits_{\mathrm{supp}(\varphi_j^1)} (\mathcal{T}_{\mathbf{y}}\mathbf{U})(\mathbf{x}_i,\mathbf{y})\varphi_j^1(\mathbf{y})\mathrm{d}s_{\mathbf{y}}\mathbf{v}_j + (\mathcal{C}(\mathbf{x}_i) - \mathbf{I})\,\mathbf{v}(\mathbf{x}_i) \,. \tag{19}$$

Accordingly, the discretized single and double layer integral operator are defined by

$$\mathsf{V}_{ij} \coloneqq \int\limits_{\mathrm{supp}(\varphi_j^0)} \mathbf{U}(\mathbf{x}_i,\mathbf{y})\varphi_j^0(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \tag{20}$$

and

$$\mathsf{K}_{ij} \coloneqq \fint\limits_{\mathrm{supp}(\varphi_j^1)} (\mathcal{T}_{\mathbf{y}}\mathbf{U})(\mathbf{x}_i,\mathbf{y})\varphi_j^1(\mathbf{y})\mathrm{d}s_{\mathbf{y}} \,, \tag{21}$$

respectively. The fundamental solutions are three-by-three tensors and thus for each index i and j the operators $\mathsf{V}_{ij}, \mathsf{K}_{ij} \in \mathbb{K}^{3\times 3}$.

Finally, the discretization of the boundary integral equations lead in general to the linear system of the form

$$\mathcal{A}\,\mathbf{y} = \mathbf{f} \,, \quad \mathcal{A} \in \mathbb{K}^{N\times N} \,, \quad \mathbf{y}, \mathbf{f} \in \mathbb{K}^N \,, \tag{22}$$

where N denotes the number of degrees of freedom (DOFs) and one finds $N = 3N_p$, with N_p the number of collocation points. The discrete linear system of the direct approach is achieved through rearranging the boundary data. The known data are assembled in $\mathbf{f}$ and the unknowns in $\mathbf{y}$. It may be noted that the double layer operator is regularized with partial integration. This approach can be found for elastostatics in Ref. [28], which can be transferred directly to elastodynamics as presented in Ref. [30]. As a result, weakly singular integral equations are obtained. The weakly singular integrals are treated with the formulas of Erichsen and Sauter [17] and all regular integrals with the standard Gauss quadrature. In the case of a mixed boundary

value problem, one gains an equation as in (22) but then $\mathcal{A}$ is divided into parts that are related to those of the SLP and those of the DLP. A Schur complement based solver is then used (see, e.g. Ref. [30]). Nevertheless, the following discussion on ACA applies analogously to the submatrices of $\mathcal{A}$.

3 Adaptive Cross Approximation

As discussed in the previous section, the boundary element discretization of boundary value problems leads to large fully populated matrices $\mathcal{A} \in \mathbb{K}^{N\times N}$. Methods like fast multipole [22, 38] or panel clustering [27] generates a low-rank approximation by replacing the kernel function of the integral operator. In contrast to these methods the singular value decomposition (SVD) and the adaptive cross approximation (ACA) [2, 5] operates purely algebraic. A low-rank approximation of matrix $\mathrm{A} \in \mathbb{K}^{m\times n}$ with rank k and $k(m+n) \ll mn$ is defined to be

$$\mathrm{A} \approx \mathrm{S}_k = \mathrm{U}\,\mathrm{V}^H \quad \text{with } \mathrm{U} \in \mathbb{K}^{m\times k} \text{ and } \mathrm{V} \in \mathbb{K}^{n\times k}\,. \tag{23}$$

Hence, the memory requirement is reduced from $O(mn)$ to $O(k(m+n))$. The ACA algorithm uses only few of the original matrix entries to compute the low-rank matrix. Compared to the SVD, which would find the lowest rank for a given accuracy, it is not necessary to compute the whole matrix beforehand. The algorithm will detect adaptively which rows and columns have to be computed. Incidentally, the computational complexity of order $O(N^3)$ makes the SVD unattractive. In contrast, the ACA algorithm is less expensive and very powerful. The only condition of the algorithm is that the discrete integral operators have sufficiently smooth kernels.

3.1 Clustering

The first step prior to approximate $\mathcal{A}$ by use of the adaptive cross approximation is to structure the computational domain. For this purpose the hierarchical matrices or $\mathcal{H}$-matrices are used. We refer the reader to Refs. [3, 21, 25] for a deeper discussion of this subject. The concept basically is to decompose the matrix into subblocks first and then perform a low-rank approximation to suitable subblocks. Relating to the first point, the partitioning of the matrix is conducted by a recursive subdivision of the geometry. This implies a decomposition of the array of degrees of freedom based on a certain strategy. The index set I_0, e.g. $I_0 = \{1, ..., N_p\}$, is subdivided into two son clusters based on the principal component analysis (PCA). Recursively performing this procedure generates a balanced cluster tree. A son cluster is not further subdivided if it contains at most a prescribed size $b_{\min}$ and is referred to as leaf. After creating the cluster tree, the block cluster tree or the hierarchical structure of the matrix is constructed with the aid of the distance criterion

$$\min\{\mathrm{diam}(Cl_x), \mathrm{diam}(Cl_y)\} \leq \eta\, \mathrm{dist}(Cl_x, Cl_y) \tag{24}$$

with a given parameter $\eta \in \mathbb{R}^+$. The corresponding index set to cluster Cl_x is denoted by I and to cluster Cl_y by J. In this way, the indices of the matrix are permuted in such a way that a hierarchy of blocks arises, which are classified as far-field or near-field. A subblock of the matrix $\mathrm{A} := \mathcal{A}_{I\times J}$ is admissible and is called far-field if the criterion (24) is fulfilled. Subsequently, we concentrate only on admissible blocks of the hierarchical matrix and approximate them by

the ACA. If a cluster pair does not satisfy condition (24), then the entries of the corresponding matrix block are computed by the standard boundary element method and stored without approximation. In this case, the block is classified as near-field.

3.2 Matrix-valued ACA algorithm

The partial differential equations of elastostatics (6) or elastodynamics (1) are equations with vector-valued unknowns. Therefore, the corresponding system matrix is constructed by submatrices of size 3×3. Each pair of nodes on the mesh define one submatrix. To make use of the ACA on this kind of problems the conventional algorithm has to be partially modified. In the work of Messner and Schanz [34] the ACA has been applied to elastodynamics. In this study, the matrix-valued problem has been decomposed in each direction. As a consequence, the system matrix has been partitioned into nine subproblems and, afterwards, each subproblem has been approximated with conventional ACA independently. Our intention is to extend the ACA algorithm to avoid this reordering of the system matrix and to use the conventional ACA algorithm straightforward. The essential component is to define a suitable pivot element.

Before we discuss the ACA algorithm for vector-valued problems in more detail we briefly recap the conventional ACA introduced by Bebendorf and Rjasanow [2, 5]. In the literature two different approaches of the algorithm are known, the fully-pivoted and the partially-pivoted ACA. The first mentioned approach computes the approximation S_k and residual matrix R_k at each iteration and defines the pivot as the largest entry in the residual matrix. The matrix $\mathrm{A} \in \mathbb{K}^{m \times n}$ given by $\mathrm{A} = \mathrm{S}_k + \mathrm{R}_k$ at each iteration have first to be fully computed and afterwards the residual matrix is minimized. Contrary to this, the partially-pivoted approach assembles only one row $\mathrm{A}[i_k,:]$ and one column $\mathrm{A}[:,j_k]$ of the matrix A at each iteration. For $k = 0,1,2,\ldots$ the rows and columns of the matrix approximant $\mathrm{S}_k = \sum_{m=1}^{k} \mathbf{u}_m \mathbf{v}_m^H$ are computed by the steps given in Algorithm 1. A full explanation of this algorithm can be found in Rjasanow and Steinbach [36]. The algorithm starts with an arbitrarily chosen row index i_k. The pivot element γ_{k+1} is defined

Algorithm 1 Conventional ACA

1: **repeat**
2: row of the residual $\mathbf{r}_\mathrm{v} \coloneqq (\mathrm{A}[i_k,:])^H - \sum_{m=1}^{k} (\mathbf{u}_m)_{i_k} \mathbf{v}_m$
3: find the column index $j_k \coloneqq \operatorname{argmax}_j |(\mathbf{r}_\mathrm{v})_j|$
4: compute pivot $\gamma_{k+1} \coloneqq (\mathbf{r}_\mathrm{v})_{j_k}^{-1}$
5: column of the residual $\mathbf{r}_\mathrm{u} \coloneqq \mathrm{A}[:,j_k] - \sum_{m=1}^{k} (\mathbf{v}_m)_{j_k} \mathbf{u}_m$
6: compute low-rank vectors $\mathbf{u}_{k+1} \coloneqq \mathbf{r}_\mathrm{u}$ $\mathbf{v}_{k+1} \coloneqq \gamma_{k+1} \mathbf{r}_\mathrm{v}$
7: **until** stopping criterion (25) is fulfilled

by the inverse of $(\mathbf{r}_\mathrm{v})_{j_k}$, where the column index j_k is selected as the maximum entry in modulus of the residual row $\mathbf{r}_\mathrm{v}$. Likewise, the row index for the next iteration is fixed by the maximum entry in modulus of the residual column $\mathbf{r}_\mathrm{u}$. The iteration of Algorithm 1 stops if the following

criterion

$$\|\mathbf{u}_{k+1}\|_F \|\mathbf{v}_{k+1}\|_F \leq \varepsilon_{\mathrm{ACA}} \|\mathrm{S}_{k+1}\|_F \tag{25}$$

with a prescribed accuracy $\varepsilon_{\mathrm{ACA}}$ is satisfied. The Frobenius norm of the approximation $\mathrm{S}_{k+1} \coloneqq \mathrm{S}_k + \mathbf{u}_{k+1}\mathbf{v}_{k+1}^H$ is defined by

$$\|\mathrm{S}_{k+1}\|_F^2 = \|\mathrm{S}_k\|_F^2 + 2\Re\langle \mathrm{S}_k, \mathbf{u}_{k+1} \otimes \mathbf{v}_{k+1} \rangle + \|\mathbf{u}_{k+1}\|_F^2 \|\mathbf{v}_{k+1}\|_F^2 \ . \tag{26}$$

Finally, the approximant of A is given by

$$\mathrm{S}_k = \sum_{m=1}^{k} \mathbf{u}_m \mathbf{v}_m^H \tag{27}$$

and possess the low-rank order k. In this paper, only the partially-pivoted ACA is considered.

Now, let's focus on the ACA algorithm for vector-valued problems. In section 2, we have explained and in particular specified in (20) and (21) that the system matrix after discretization of the boundary integral equation is essentially constructed as

$$\mathcal{A}[i,j] = \int\limits_{\mathrm{supp}(\varphi_j)} \mathrm{K}(\mathbf{x}_i, \mathbf{y}) \varphi_j(\mathbf{y}) \mathrm{d}s_{\mathbf{y}} \ , \tag{28}$$

where the kernel function $\mathrm{K} : \Gamma \times \Gamma \to \mathbb{K}^{3\times 3}$ generates a three-by-three matrix. The index i corresponds to one collocation point and j to one basis function. As we mentioned above, the crucial part of the extension of the algorithm is to define a suitable pivot element. The pivot element is now a matrix of size 3×3. From this point of view, the strategy has to ensure the regularity of the pivot matrix. One possibility to choose the pivot matrix is to consider the singular values of every 3×3 block of the residual block row or residual block column. This strategy has been discussed and elaborated in Refs. [37] and [13]. In Algorithm 2 the matrix-valued ACA is outlined. We consider again an admissible block $\mathrm{A} \in \mathbb{K}^{m\times n}$, where $m = 3|I|$ and $n = 3|\mathcal{J}|$, respectively. The cardinality of the row $|I|$ or column $|\mathcal{J}|$ index set is designated by the number of points in the corresponding cluster. Therefore, the row index i or column index j actually corresponds to a block row index or block column index, respectively. As a result, the row or column of the residual are not vectors anymore they are type of block row vectors $\mathbf{v} \in \mathbb{K}^{3\times n}$ or block column vectors $\mathbf{u} \in \mathbb{K}^{m\times 3}$. The first block row index i_k is chosen again arbitrarily. The generation of the block row $\mathrm{A}[i_k, :]$ is of size $3 \times n$ and the size of the block column $\mathrm{A}[:, j_k]$ is $m \times 3$. In addition, the approximants are: $\mathbf{U}$ is an $(m \times 3k)$-matrix and $\mathbf{V}$ is a $(3k \times n)$-matrix in $\mathbb{K}$. The stopping criterion reads as

$$\|\mathbf{u}\|_F \|\mathbf{v}\|_F \leq \varepsilon_{\mathrm{ACA}} \|\mathbf{S}_{k+1}\|_F \tag{29}$$

with the Frobenius norm of the current approximation

$$\|\mathbf{S}_{k+1}\|_F^2 = \|\mathbf{S}_k + \mathbf{u}\,\mathbf{v}\|_F^2 = \|\mathbf{S}_k\|_F^2 + 2\Re\langle \mathbf{S}_k, \mathbf{u}\,\mathbf{v} \rangle_F + \|\mathbf{u}\|_F^2 \|\mathbf{v}\|_F^2 \ . \tag{30}$$

The pivot position is determined by computing the minimal singular value $\min\sigma(\widetilde{\mathbf{v}}[:, j])$ of every block of the residual row $\widetilde{\mathbf{v}}$ and then taking the maximum of it

$$j_k = \underset{j}{\mathrm{argmax}} \left\{ \min\sigma(\widetilde{\mathbf{v}}[:, j]) \right\} \ . \tag{31}$$

Algorithm 2 matrix-valued ACA

1: Initialization
2: **repeat**
3: row of the residual $\widetilde{\mathbf{v}} \coloneqq \mathrm{A}[i_k,:] - \mathbf{U}[i_k,:]\,\mathbf{V}$
4: pivot position $j_k := \mathrm{PivotPosition}(\widetilde{\mathbf{v}})$
5: column of the residual $\mathbf{u} \coloneqq \mathrm{A}[:,j_k] - \mathbf{U}\,\mathbf{V}[:,j_k]$
6: update
$\mathbf{v} \coloneqq (\mathrm{A}[i_k,j_k])^{-1}\,\widetilde{\mathbf{v}}$
$\mathbf{U} \coloneqq \big[\mathbf{U},\mathbf{u}\big], \quad \mathbf{V} \coloneqq \begin{bmatrix}\mathbf{V}\\ \mathbf{v}\end{bmatrix}$
7: **until** stopping criterion (29) is fulfilled

The singular values of a matrix Q are the square roots of the eigenvalues of the Hermitian and positive semidefinite matrix $Q^H Q$ and, therefore, the singular values are real and non-negative. The same strategy is used to define the next block row index. The low-rank approximation of the matrix A with order $K = 3k$ is now given by

$$\mathrm{A} \approx \mathrm{S}_K = \mathbf{U}\mathbf{V}\,. \tag{32}$$

Finally, a compressed matrix block is obtained by applying this introduced matrix-valued ACA until the stopping criterion is fulfilled. The low-rank approximation of all admissible blocks then gives the approximation of the $\mathcal{H}$-matrix of the considered problem.

4 Numerical examples

In this section, we present numerical experiments to validate the proposed partially-pivoted matrix-valued ACA algorithm. First, the indirect SLP and DLP approach of a Dirichlet boundary value problem and, secondly, the direct approach of a mixed problem for elastostatics (6) and elastodynamics (1) are analyzed. All computations were executed with material parameters

$$E = 1\,\mathrm{N/m^2}\,,\ \nu = 0.2 \tag{33}$$

in terms of elastostatic problems and with

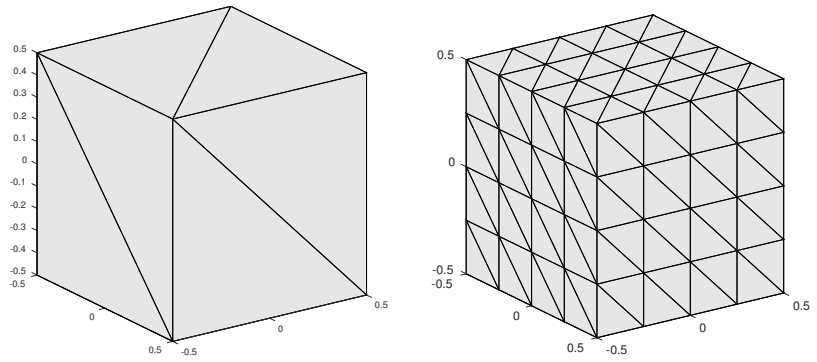
$$E = 1\,\mathrm{N/m^2}\,,\ \nu = 0\,,\ \rho = 1\,\mathrm{kg/m^3} \tag{34}$$

relating to elastodynamic problems. The latter yields to a compression wave velocity of $c_P = 1\,\mathrm{m/s}$ and to a shear wave velocity of $c_S = \sqrt{1/2}\,\mathrm{m/s}$. Moreover, the complex frequency is chosen to be $s = (1+\mathrm{i})\,\mathrm{s}^{-1}$. For a study on the behavior with respect to frequency see Ref. [13]. The admissibility parameter η of the distance criterion (24) is set to $\eta = 0.8$ in all examples. For the leaf size of the clusters the value of $b_{\min} = 40$ is used.

All numerical examples have been computed with HyENA, a C++ based BEM-library developed at the Institute of Applied Mechanics at Graz University of Technology [33].

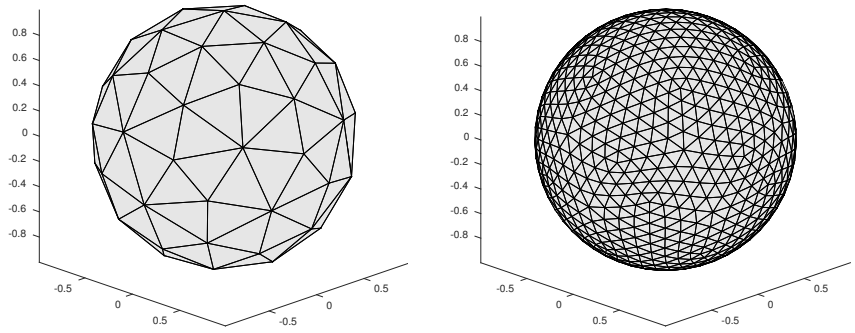
4.1 Computational domains

For the convergence results, the considered computational domains are on the one hand the unit cube and on the other hand the unit sphere. Both are centered at the origin. Furthermore, a sequence of triangulations by uniform refinement of the initial mesh (level $\ell = 0$) is generated. The discretization of the geometries at different refinement levels ℓ are shown in Fig. 1 and Fig. 2.



(a) $\ell = 0$ (b) $\ell = 2$

Figure 1: Discretizations of the unit cube (N=12) and its refinement (N=192)



(a) $\ell = 0$ (b) $\ell = 2$

Figure 2: Discretizations of the unit sphere (N=122) and its refinement (N=1952)

One final note, the resulting linear system is solved by GMRES with a specified accuracy of $\varepsilon_{\text{GMRES}} = 10^{-8}$.

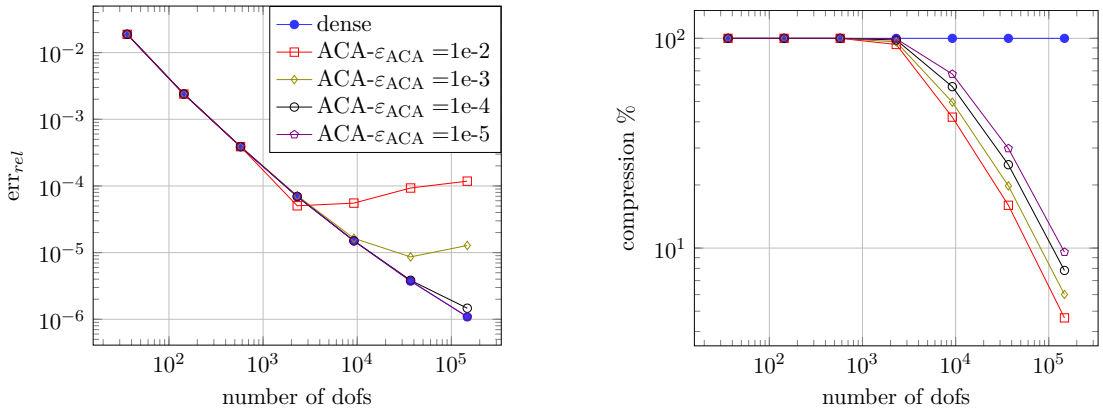
4.2 Indirect problems

The accuracy of the approximate solution $\mathbf{u}_h(\mathbf{x})$ is measured by the relative pointwise error at some sampling points. We choose as analytic solution the fundamental solution $\mathbf{u}(\mathbf{x}) = \mathbf{U}(\mathbf{x}, \mathbf{y}^*)$

with the source point located at $\mathbf{y}^* = (1,1,1)^\top$. Accordingly, the error measure is defined as

$$\mathrm{err}_{rel}^{\ell} = \frac{\sqrt{\sum_{i=1}^{N_{pts}} \left(\mathbf{u}(\mathbf{x}_i) - \mathbf{u}_{h_\ell}(\mathbf{x}_i)\right)^2}}{\sqrt{\sum_{i=1}^{N_{pts}} \mathbf{u}^2(\mathbf{x}_i)}}, \tag{35}$$

with the number of interior points N_{pts} uniformly distributed close to the origin. The verification of the matrix-valued ACA is done by comparing the relative error with the computation without any low-rank approximation (dense) on the one hand and by comparing the compression rate with the SVD on the other hand. The compression is defined as the ratio of the amount of memory consumed for the low rank approximation to the amount of memory required for a dense matrix. One of the essential parameters is the low-rank accuracy $\varepsilon_{\mathrm{ACA}}$. In a first comparison, the relative error and the compression rate of the indirect SLP approach in elastostatics are presented in Fig. 3. The chosen geometry here is the unit cube. The expected result can be



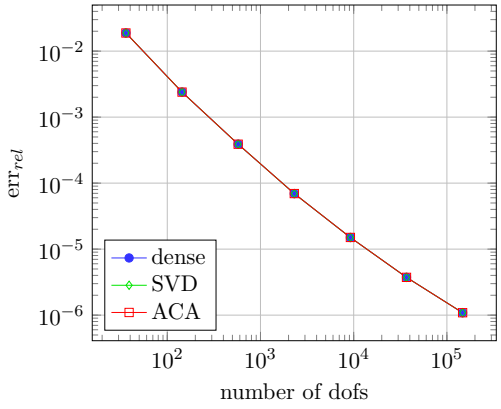
(a) Error (b) Compression

Figure 3: Indirect SLP approach for elastostatic on the unit cube while keeping the low-rank accuracy constant

observed. Up to a certain error level the ACA based computation has the same error as the dense computation, i.e. the discretization error is greater than the matrix approximation error. However, from a distinct point the low-rank approximation error dominates and the relative error stagnates. Fig. 3b clearly shows that the lower the approximation quality is chosen, the better the compression can be. Nevertheless, a remarkable compression can be achieved even at low error levels.

Certainly, the low-rank accuracy can be adjusted in each spatial refinement level such that the discretization error is dominating, i.e. the order of convergence of the dense computation is maintained. Such a study is provided in Fig. 4.

In this paper, the approximation of the ACA next to the SVD is considered. The low-rank accuracy for the SVD is denoted with $\varepsilon_{\mathrm{SVD}}$. The necessary accuracies are listed in the table



ℓ	number of dofs	$\varepsilon_{\mathrm{SVD}}$	$\varepsilon_{\mathrm{ACA}}$
3	2304	1e-4	1e-4
4	9216	1e-5	1e-5
5	36864	1e-6	1e-6
6	147456	1e-7	1e-7

Figure 4: Error for the indirect SLP approach on the unit cube with prescribed accuracy of ACA/SVD

in Fig. 4 on the right side. As expected, it is necessary to increase the low-rank accuracy with the refinement of the spatial meshes. However, the error shown on the left side of Fig. 4 is then the same for the dense computation and for the computation with either ACA or SVD approximation. In the right table solely the accuracies of the refinement levels are illustrated in which compression was achieved. At each level, the accuracy of SVD and ACA was chosen identically.

Since the SVD leads to the best low-rank approximation, we compare the compression rate of the matrix-valued ACA with it. Examples of the compression rates of the indirect SLP are depicted in Fig. 5. In these examples the problem of elastostatics and elastodynamics for the

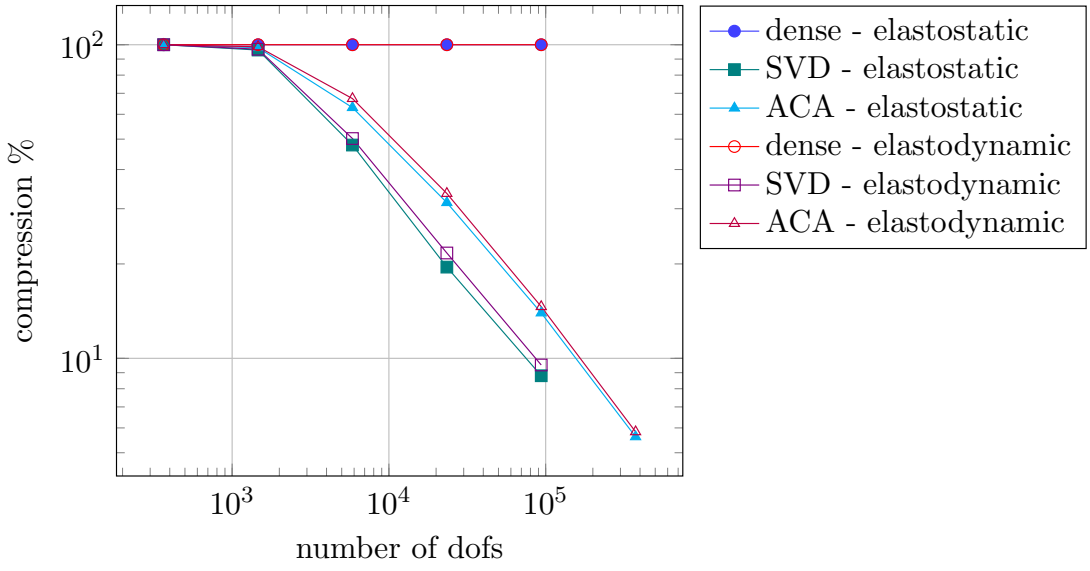


Figure 5: Compression of the indirect SLP approach for the unit sphere

unit sphere are examined. The results indicate that the ACA attains a sufficiently good low-rank order. At the finest refinement level the compression is well below 10 %. Because of this, we are able to compute one more refinement compared to the dense computation. Analogous results are observed in all other examples.

Even though these results are very promising, the matrix-valued ACA Algorithm 2 has limits. The approximation of a block fails if all three-by-three matrix entries in $\mathbf{v}$ are singular. In particular, the double layer operator of elastostatics fails for the unit cube. In essence, the main diagonal of the DLP kernel has a normal derivative of the distance vector. Therefore, the traction fundamental solution $(\mathcal{T}_{\mathbf{y}}\mathbf{U})(\mathbf{x},\mathbf{y})$ becomes singular, if the points $\mathbf{x}$ and $\mathbf{y}$ are on the same plane. If this happens for all entries in $\mathbf{v}$ the algorithm will fail (see also the respective remark in Ref. [13]). Nevertheless, it is evident and presented in Fig. 6 that the approximation of the DLP for the unit sphere by the matrix-valued ACA performs well.

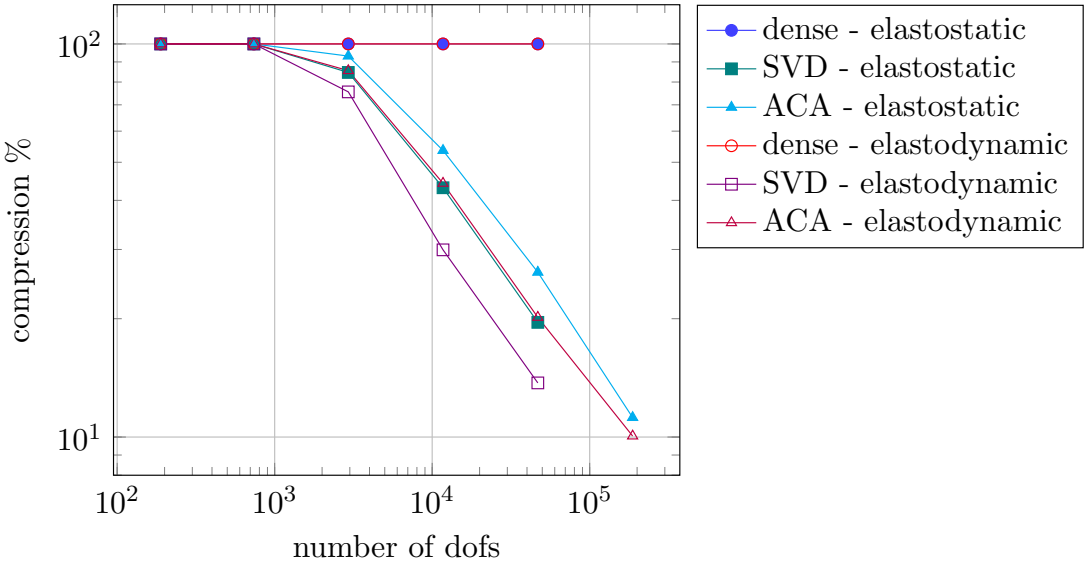


Figure 6: Compression of the indirect DLP approach for the unit sphere

The compression rates presented for the DLP in Fig. 6 are comparable to the results shown above and for the finest mesh below 10%. It should be noted that also in this study the low-rank approximation is adjusted to the refinement levels so that the relative inner pointwise error of the dense computation is maintained. For engineering computations an error level of 10^{-8} as in the tests above is usually not necessary. Hence, much better compression rates can be obtained by decreasing the low-rank accuracy.

A different representation of the efficiency of an ACA based BEM results from the consideration of the absolute storage values. In Fig. 7, the storage requirements against the number of dofs are plotted. As example the SLP for the unit sphere is taken, i.e. the storage requirements from the test presented in Fig. 5 are shown. Obviously, the storage requirement of the dense computation grows with the expected quadratic order. On the contrary, the presented ACA based computation shows an almost linear behavior, i.e. approximately the order $O\left(N\log^4(N)\right)$. The logarithmic term can also be observed for scalar-valued problems but not with a power of four.

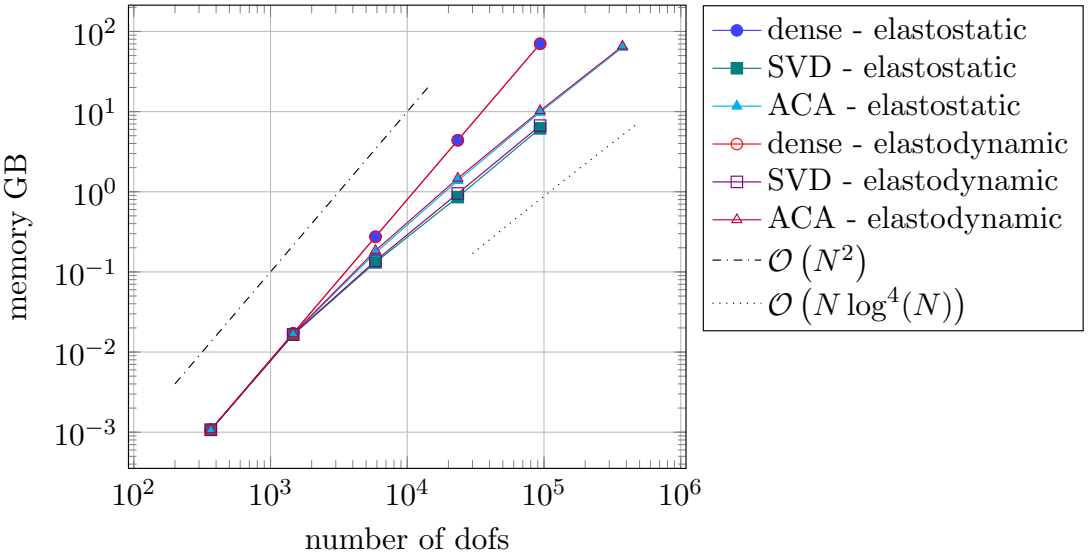


Figure 7: Storage requirements of the indirect SLP approach for the unit sphere

However, also FMM formulations for vector-valued problems show this logarithmic term in the complexity. The interpolation based FMM for elastodynamics have, e.g. a power of six on the logarithmic term [39]. It may be remarked that in other studies a lower order of the logarithmic term is reported. In these cases, the low-rank accuracy is kept constant during refinement.

Finally, the computation time of a matrix-vector product of the low-rank approximated matrix is compared to the standard BEM matrix, see Fig. 8. The data for indirect SLP approach for the unit sphere are used. From about 10^4 number of dofs on, the multiplication of the low-rank approximation becomes much faster. Certainly, the total CPU-time depends heavily on programming, compiler, hardware, etc., but the tendency will be the same. This acceleration of the matrix-vector product of the ACA based BEM has a tremendous impact in the solution time due to the repeated application in an iterative solver like GMRES.

4.3 Mixed problem

To show the behavior for mixed boundary value problems, the unit sphere is considered as computational domain. Dirichlet boundary conditions are imposed on one hemisphere and Neumann boundary conditions on the other. As before, the Dirichlet data are given by the displacement fundamental solution with a source point outside of the domain. Accordingly, the Neumann data are attained by applying the traction operator on the displacement fundamental solution. The error of both, the displacement and the traction field, is measured in the L_2-norm on the boundary. This means that the displacement error is given by

$$\mathrm{err}_{L_2}^{\ell}(\mathbf{u}) = \|\mathbf{u}-\mathbf{u}_{h_\ell}\|_{L_2(\Gamma)} = \left(\sum_{i=1}^{N_n} \int_{\mathrm{supp}(\varphi_i^1)} \left(\mathbf{u}(\mathbf{x}) - \mathbf{u}^{\ell}(\mathbf{x}_i)\varphi_i^1(\mathbf{x})\right)^2 \mathrm{d}s_{\mathbf{x}} \right)^{\frac{1}{2}}$$

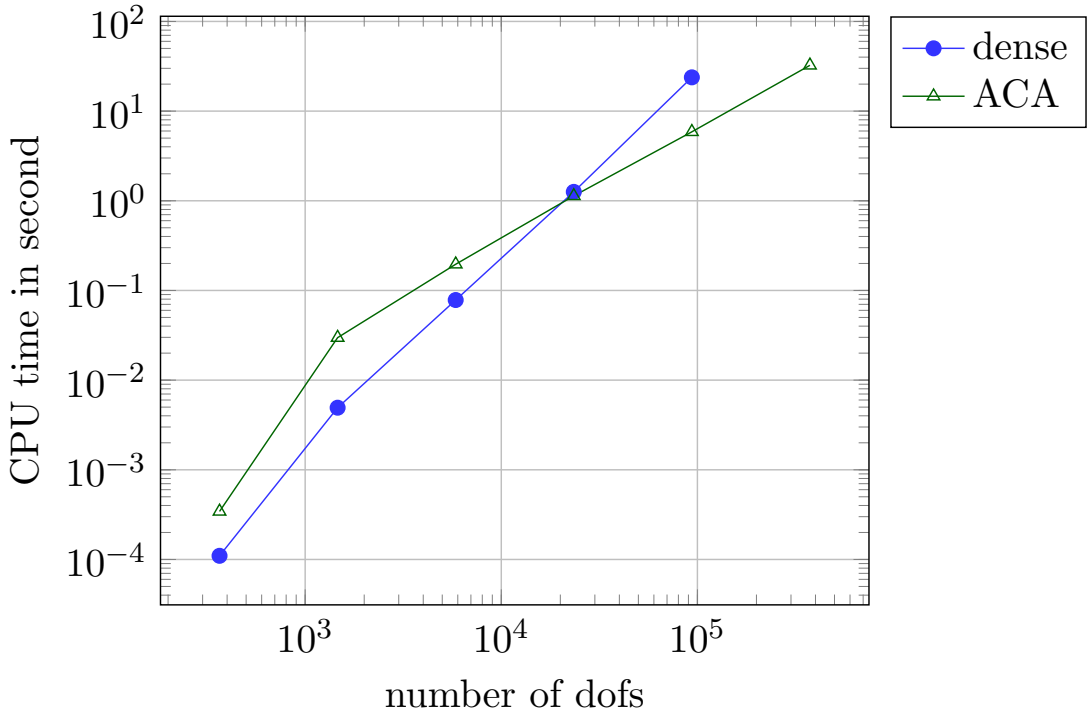


Figure 8: Computation time of a matrix-vector product (SLP)

and the traction error is determined by

$$\mathrm{err}_{L_2}^{\ell}(\mathbf{t}) = \|\mathbf{t}-\mathbf{t}_{h_\ell}\|_{L_2(\Gamma)} = \left(\sum_{i=1}^{N_e} \int_{\mathrm{supp}(\varphi_i^0)} \left(\mathbf{t}(\mathbf{x}) - \mathbf{t}^{\ell}(\mathbf{x}_i)\varphi_i^0(\mathbf{x}) \right)^2 \mathrm{d}s_{\mathbf{x}} \right)^{\frac{1}{2}} .$$

For mixed problems, the low-rank approximation is examined only by the ACA. The admissibility parameter η for the distance criterion is as well chosen to be 0.8, and the low-rank accuracy $\varepsilon_{\mathrm{ACA}}$ is set such that the convergence order in the L_2-norm is comparable to the dense computation. As with the indirect approach, this setting results in an increasing of the low-rank accuracy at each refinement step. The cluster parameter $\mathrm{b}_{\mathrm{min}}$ was set to 40 as before in all examples.

Fig. 9 shows the order of convergence. As anticipated, the Dirichlet data exhibit a quadratic order and the Neumann data a linear order.

The findings of the compressions of both problems, elastostatics and elastodynamics, are displayed in Fig. 10.

Again, the matrix-valued ACA obtains an excellent compression rate while maintaining the quality of the solution. Essentially, there are no differences to the indirect approach.

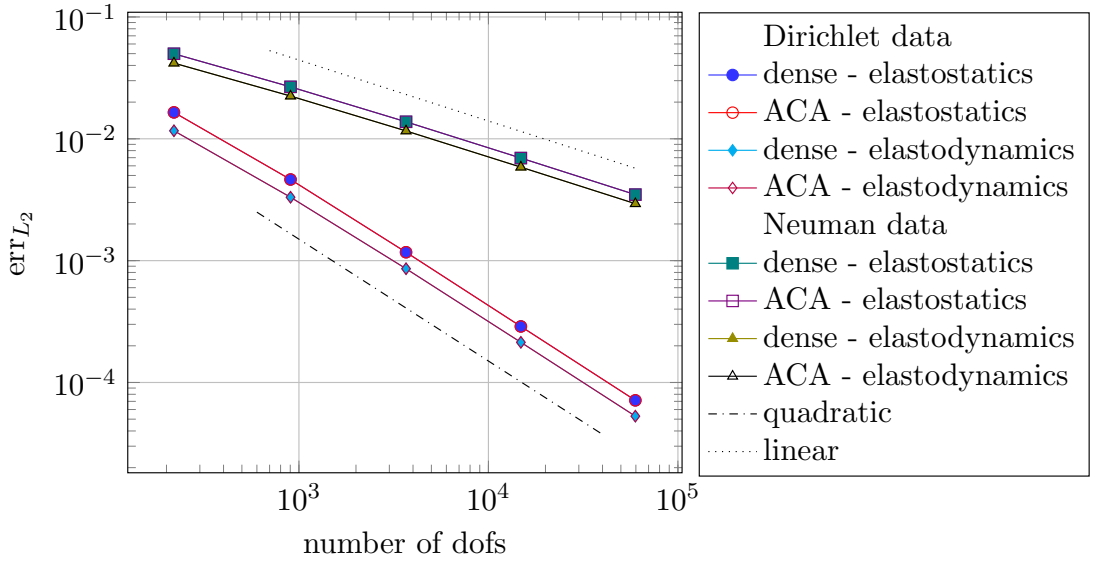


Figure 9: L_2-error of the mixed problem for the unit sphere

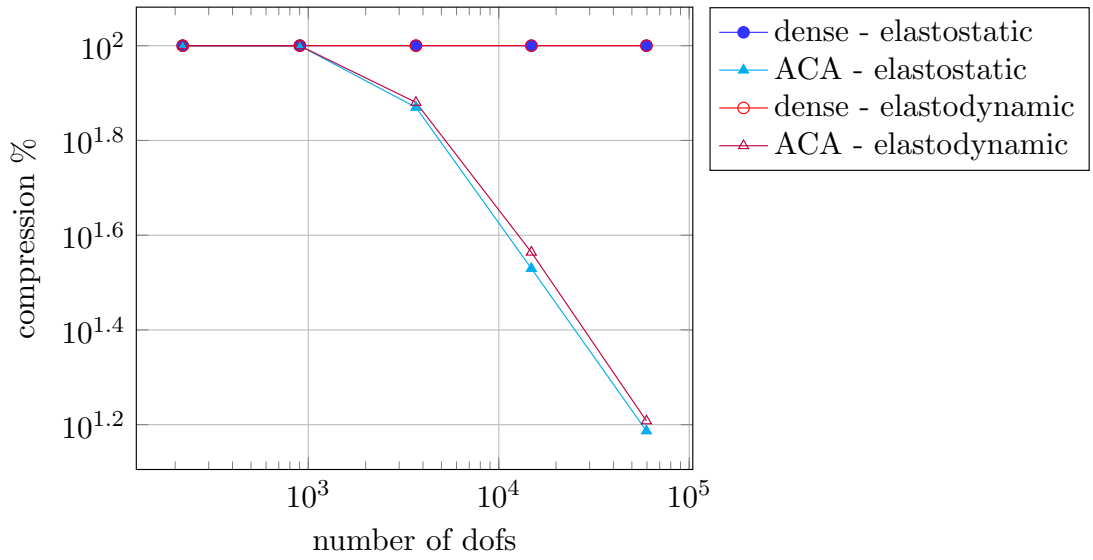


Figure 10: Compression of mixed boundary value problems for the unit sphere

5 Conclusions

The boundary element method for elastostatics and its dynamic counterpart is well established. In the beginning of the paper the respective partial differential equations and their related integral equations have been recalled. The fully populated system matrices of the discrete integral operators result in an overall quadratic complexity with respect to storage and computing time. To reduce this complexity so-called fast methods have been developed. Here, the adaptive cross

approximation (ACA) is used to obtain an almost linear complexity. The essential point of ACA is to find a suitable pivot strategy, where for vector-valued problems like elastostatics and elastodynamics this pivot element is a three-by-three matrix resulting from the fundamental solutions. Following a suggestion of Rjasanow and Weggler the singular values of these three-by-three matrices are considered. The smallest singular value of each block of fundamental solutions are compared and their maximal value determines the pivot three-by-three matrix. Using this strategy, a robust algorithm can be defined. Numerical studies have shown that the algorithm preserves the rate of convergence of a classical (dense) formulation. The obtained compression rates are satisfactory. In the presented examples the compression has been smaller than 10% in spite of a required precision of 10^{-8}. For real world engineering applications much lower low-rank accuracies are acceptable due to the errors introduced from the mechanical model and thus, much better compression rates are possible.

It should be noted that the algorithm has difficulties with the double layer operator for large plane surfaces. If all corresponding points are on the same plane whitin an admissible block, the determination of the pivot element may fail.

Fortunately, this situation is in real world applications very rare. However, the problem may be overcome by certain cluster strategies which avoid this specific case.

References

[1] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solutions of second-kind integral equations. *SIAM J. Sci. Comput.*, 14(1):159–184, 1993.

[2] M. Bebendorf. Approximation of boundary element matrices. *Num. Math.*, 86(4):565–589, Oct 2000.

[3] M. Bebendorf. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*. Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, 2008.

[4] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. *Math. Meth. Appl. Sci.*, 29(14):1721–1747, 2006.

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

[6] I. Benedetti, M. H. Aliabadi, and G. Davì. A fast 3D dual boundary element method based on hierarchical matrices. *Int. J. Solids Structures*, 45(7–8):2355–2376, 2008.

[7] D. E. Beskos. Boundary element methods in dynamic analysis. *AMR*, 40(1):1–23, 1987.

[8] D. E. Beskos. Boundary element methods in dynamic analysis: Part II (1986-1996). *AMR*, 50(3):149–197, 1997.

[9] M. Bonnet. *Boundary Integral Equation Methods for Solids and Fluids*. John Wiley & Sons, 1999.

[10] S. Börm and L. Grasedyck. Hybrid cross approximation of integral operators. *Num. Math.*, 101(2):221–249, Aug 2005.

[11] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. *Boundary Element Techniques*. Springer-Verlag, Berlin, New York, 1984.

[12] S. Chaillat, M. Bonnet, and J.-F. Semblat. A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain. *Comput. Methods Appl. Mech. Engrg.*, 197(49–50):4233–4249, 2008.

[13] S. Chaillat, L. Desiderio, and P. Ciarlet. Theory and implementation of h-matrix based iterative and direct solvers for helmholtz and elastodynamic oscillatory kernels. *J. Comput. Phys.*, 351(Supplement C):165 – 186, 2017.

[14] M. Costabel. Time-dependent problems with the boundary integral equation method. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, Fundamentals, chapter 25, pages 703–721. John Wiley & Sons, New York, Chichester, Weinheim, 2005.

[15] T. Cruse and F. Rizzo. A direct formulation and numerical solution of the general transient elastodynamic problem. i. *J. Math. Anal. Appl.*, 22(1):244 – 259, 1968.

[16] J. Domínguez. *Boundary Elements in Dynamics*. Computational Mechanics Publication, Southampton, 1993.

[17] S. Erichsen and S. A. Sauter. Efficient automatic quadrature in 3-d Galerkin BEM. *Comput. Methods Appl. Mech. Engrg.*, 157(3–4):215–224, 1998.

[18] W. Fong and E. Darve. The black-box fast multipole method. *J. Comput. Phys.*, 228(23):8712–8725, 2009.

[19] L. Gaul, M. Kögl, and M. Wagner. *Boundary Element Methods for Engineers and Scientists*. Springer-Verlag Berlin Heidelberg, 2003.

[20] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Lin. Algebra Appl.*, 261(1–3):1–21, 1997.

[21] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, Aug 2003.

[22] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325 – 348, 1987.

[23] L. Greengard and V. Rokhlin. A new version of the Fast Multipole Method for the Laplace equation in three dimensions. *Acta Num.*, 6:229–269, 1997.

[24] T. Grytsenko and A. N. Galybin. Numerical analysis of multi-crack large-scale plane problems with adaptive cross approximation and hierarchical matrices. *Eng. Anal. Bound. Elem.*, 34(5):501–510, 2010.

[25] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, Apr 1999.

[26] W. Hackbusch and S. Börm. Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices. *Computing*, 69(1):1–35, Sep 2002.

[27] W. Hackbusch and Z. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54(4):463–492, 1989.

[28] H. Han. The boundary integro-differential equations of three-dimensional Neumann problem in linear elasticity. *Numer. Math.*, 68(2):269–281, 1994.

[29] F. Hartmann. The somigliana identity on piecewise smooth surfaces. *J. Elasticity*, 11:403–423, 1981.

[30] L. Kielhorn and M. Schanz. Convolution quadrature method based symmetric Galerkin boundary element method for 3-d elastodynamics. *Int. J. Numer. Methods. Engrg.*, 76(11):1724–1746, 2008.

[31] F. Maerten. Adaptive cross-approximation applied to the solution of system of equations and post-processing for 3D elastostatic problems using the boundary element method. *Eng. Anal. Bound. Elem.*, 34(5):483–491, 2010.

[32] V. Mantič. A new formula for the c-matrix in the somigliana identity. *J. Elasticity*, 33(3):191–201, Dec 1993.

[33] M. Messner, M. Messner, F. Rammerstorfer, and P. Urthaler. Hyperbolic and elliptic numerical analysis BEM library (HyENA). `http://www.mech.tugraz.at/HyENA`, 2010. [Online; accessed 22-January-2010].

[34] M. Messner and M. Schanz. An accelerated symmetric time-domain boundary element formulation for elasticity. *Eng. Anal. Bound. Elem.*, 34(11):944 – 955, 2010. Special issue on the advances in mesh reduction methods- In honor of Professor Subrata Mukherjee on the occasion of his 65th birthday.

[35] 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.

[36] S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations (Mathematical and Analytical Techniques with Applications to Engineering)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[37] S. Rjasanow and L. Weggler. Matrix valued adaptive cross approximation. *Math. Methods Appl. Sci.*, 40(7):2522–2531, 2017.

[38] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.*, 60(2):187 – 207, 1985.

[39] T. Traub. *A Kernel Interpolation Based Fast Multipole Method for Elastodynamic Problems*, volume 29 of *Computation in Engineering and Science*. Verlag der Technischen Universität Graz, 2016.