Appl. Numer. Math. 201 (2024) 1–19

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Research Paper

Adaptive $\mathcal{H}$-matrix computations in linear elasticity

Maximilian Bauer *, Mario Bebendorf

Faculty of Mathematics, Physics and Computer Sciences, University of Bayreuth, Universitätsstraße 30, Bayreuth, 95447, Germany

ARTICLE INFO

Keywords:
Matrix adaptivity
Hierarchical matrices
Linear elasticity
ACA
Error estimation

ABSTRACT

This article deals with the efficient numerical treatment of the Lamé equations. The equations of linear elasticity are considered as boundary integral equations and solved in the setting of the boundary element method (BEM). Using BEM, one is faced with the solution of a system of equations with a fully populated system matrix, which is in general very costly. In order to overcome this difficulty, adaptive and approximate algorithms based on hierarchical matrices and the adaptive cross approximation are proposed. These new methods rely on error estimators and refinement techniques known from adaptivity but are not used here to improve the mesh. We apply these new techniques to both, the efficient solution of Lamé equations and to the multiplication with given data.

1. Introduction

The scope of the algorithms presented in this article is the adaptive numerical treatment of the Lamé equations

$$-\mu\Delta u(x) - (\lambda+\mu)\text{grad div } u(x) = f(x) \quad \text{for } x \in \Omega \tag{1}$$

describing linear elasticity on a bounded domain $\Omega \subset \mathbb{R}^3$, where λ and μ denote the Lamé constants. As boundary values, both Dirichlet conditions (fixed restraints)

$$\gamma_0^{\text{int}} u(x) = g_D(x) \quad \text{for } x \in \Gamma_D \tag{2}$$

as well as Neumann conditions (free bearings)

$$\gamma_1^{\text{int}} u(x) = g_N(x) \quad \text{for } x \in \Gamma_N \tag{3}$$

are specified, where γ_0^{int}, γ_1^{int} denote the Dirichlet and the Neumann trace operator and $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$. Additionally, we assume a positive measure of the Dirichlet part, i.e. $\int_{\Gamma_D} \mathrm{d}s > 0$, in order to guarantee the existence of a unique solution of the considered problem; see [20,24,25].

The method of choice for the numerical solution of the problem described above is the Finite Element Method (FEM). The resulting stiffness matrix is sparse. However, depending on the underlying grid, the matrices can quickly become very large. In order to overcome this problem, sophisticated multigrid methods are frequently used nowadays. Nevertheless, the general problem remains that there are many elements and volume grids are not very easy to handle.

* Corresponding author.
E-mail addresses: maximilian1.bauer@uni-bayreuth.de (M. Bauer), mario.bebendorf@uni-bayreuth.de (M. Bebendorf).

https://doi.org/10.1016/j.apnum.2024.02.007
Received 11 May 2023; Received in revised form 17 November 2023; Accepted 10 February 2024

Available online 15 February 2024

Another method reformulates the Lamé equations in the volume Ω as an integral equation on the boundary $\partial\Omega$ is the Boundary Element Method (BEM). Initially, BEM was not a real alternative to FEM as the former requires the discretization of a non-local operator and thus leads to fully populated matrices. Nevertheless, depending on the situation, BEM offers many advantages. In addition to the reduction of the spatial dimension of the problem and thus of the number of degrees of freedom, in contact problems, for instance, it is usually sufficient to compute the stresses on the boundary, which can be accessed from the BEM relatively easily and usually more accurately than from FEM. Furthermore, with the advent of fast boundary element methods, the computational complexity of BEM has significantly improved.

Methods such as fast multipole method or hierarchical matrices ($\mathcal{H}$-matrices) reduce the complexity by approximating the discretized operator to such an extent that BEM represents an alternative to FEM. While the fast multipole method [16,22] was physically motivated and designed for specific problems, hierarchical matrices could be kept more general; see [17,18]. As the name hierarchical matrix suggests, this technique is based on a hierarchical partitioning of the discrete operator into suitable blocks. Each of these blocks contains a low-rank approximation to the original block entry, with the whole matrix having only a logarithmic-linear storage requirement. The re-presentations are further advantageous in connection with iterative solution methods, which involve many matrix-vector multiplications. Hierarchical matrices offer the possibility to perform fast matrix-vector multiplications of logarithmic-linear complexity. Employing only a few of the original matrix entries, the adaptive cross approximation (ACA) [4] has become quite popular to construct the low-rank approximation on suitable blocks. The number of matrix entries was further reduced by adding another level of adaptivity to ACA; see [2]. With the so-called block-adaptive cross approximation (BACA), not every block is approximated in the same way and to the same accuracy as in ACA, but only those blocks are more accurately approximated that lead to the greatest gain in accuracy of the solution. Keeping in mind that the number of degrees of freedom of BEM is usually significantly smaller than in FEM, with modern techniques a logarithmic-linear complexity can be achieved for BEM, whereas a super-linear complexity of the commonly used LU factorization applied to FEM problems cannot be avoided. Hence, at least from the complexity point of view BEM together with fast methods for non-local operators is well-suited for the efficient numerical treatment of Lamé equations for increasingly complex problems in the future.

The aim of this article is to adapt the ideas of BACA to the construction of an adaptive version of the matrix-vector multiplication. As an application one could think about the multiplication of given data with an operator as it might occur on the right hand side of a given problem.

When multiplying a partitioned matrix with a vector x, not every approximated block has the same effect on the accuracy of the result, especially if the vector to be multiplied contains large clusters of zeros. Such a situation can for instance occur when zero Dirichlet or Neumann boundary values are applied on large parts of the boundary. In order to exploit the structural differences in the vector x and to detect the best blocks, error estimators and techniques known from adaptivity are used. Note that the block-wise low-rank approximations will be successively improved without changing the hierarchical block structure or the grid.

The paper is organized as follows. Section 2 presents basic techniques for approximation using low-rank matrices. In more detail, partitioning, cluster trees, hierarchical matrices and adaptive cross approximation are briefly discussed. In Section 3 we introduce an adaptive scheme for an approximate computation of the matrix-vector multiplication. Furthermore, the convergence of the investigated method and some properties of the proposed error estimator are analyzed. Since the techniques in Sections 2 and 3 can be applied in many situations, we will first discuss a general case before moving on to the boundary integral approximation of the equations of linear elasticity in Section 4. Adapting the ideas of BACA to the case of linear elasticity, i.e. the Lamé equations, in Section 5, we are able to compute linear elasticity in a fully adaptive manner with $\mathcal{H}$-matrices. Finally, numerical examples presented in Section 6 show a performance acceleration and a storage reduction for the numerical computation of the boundary integral formulation of linear elasticity.

2. Approximation with low-rank matrices

We consider matrix blocks $A \in \mathbb{R}^{M\times N}$ corresponding to suitable subdomains $X, Y \subset \Omega$ having the representation

$$A = \Lambda_1 \mathcal{A} \Lambda_2^*$$

with a non-local linear operator $\mathcal{A}$ which depends linearly on the bivariate kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. The prototype for such an operator is

$$(\mathcal{A}v)(x) = \int_\Omega \kappa(x,y)v(y)\,\mathrm{d}\mu_y, \quad x \in \Omega,$$

where μ denotes the corresponding measure. The operators $\Lambda_1 : L^2(X) \to \mathbb{R}^M$ and $\Lambda_2 : L^2(Y) \to \mathbb{R}^N$ are assumed to be linear. The adjoint operator $\Lambda_2^* : \mathbb{R}^N \to L^2(Y)$ is defined as

$$(\Lambda_2^* z, f)_{L^2(Y)} = z^T(\Lambda_2 f), \quad z \in \mathbb{R}^N, \quad f \in L^2(Y).$$

These two operators are used to restrict κ to $X \times Y$ and apply (possibly different) discretizations. Two examples are:

1. Petrov-Galerkin method: Choosing functions ψ_i, $i = 1,\dots,M$, and φ_j, $j = 1,\dots,N$, with $\operatorname{supp}\psi_i \subset X$ and $\operatorname{supp}\varphi_j \subset Y$ results in the discretization

$$(\Lambda_1 f)_i = \int_X f(x)\psi_i(x)\,\mathrm{d}\mu_x \quad \text{and} \quad (\Lambda_2 f)_j = \int_Y f(y)\varphi_j(y)\,\mathrm{d}\mu_y.$$

2. Collocation method: Choosing points $x_i \in X$, $i = 1, \dots, M$, and functions φ_j, $j = 1, \dots, N$, with $\operatorname{supp}\varphi_j \subset Y$ leads to

$$(\Lambda_1 f)_i = f(y_i) \quad \text{and} \quad (\Lambda_2 f)_j = \int_Y f(y)\varphi_j(y)\,\mathrm{d}\mu_y$$

provided that f can be evaluated pointwise in Y.

The approximation of A with low-rank matrices can be done by approximating the bivariate function κ with a degenerate function $\tilde{\kappa}$, i.e. there are functions $u_l : X \to \mathbb{R}$ and $v_l : Y \to \mathbb{R}$, $l = 1, \dots, k$, such that

$$\kappa(x,y) \approx \tilde{\kappa}(x,y) := \sum_{l=1}^{k} u_l(x)v_l(y), \quad x \in X,\ y \in Y. \tag{4}$$

Such an approximation automatically leads to a matrix $\tilde{A}$ of rank at most k, since with

$$a_l := \Lambda_1 u_l \in \mathbb{R}^M \quad \text{and} \quad b_l := \Lambda_2 v_l \in \mathbb{R}^N, \quad l = 1, \dots, k,$$

it follows

$$\tilde{A} = \Lambda_1 \tilde{\mathcal{A}} \Lambda_2^* = \Lambda_1 \sum_{l=1}^{k} u_l b_l^T = \sum_{l=1}^{k} (\Lambda_1 u_l) b_l^T = \sum_{l=1}^{k} a_l b_l^T,$$

where $\tilde{\mathcal{A}}$ is defined by $(\tilde{\mathcal{A}}v)(x) := \int_\Omega \tilde{\kappa}(x,y)v(y)\,\mathrm{d}\mu_y$. The reversal of the statement is not true in general.

A matrix $\tilde{A} \in \mathbb{R}^{M\times N}$ having rank k is called low- rank matrix if the condition

$$k(M+N) < M \cdot N$$

is fulfilled. Using the outer product representation, i.e. $\tilde{A} = UV^T$ with matrices $U \in \mathbb{R}^{M\times k}$ and $V \in \mathbb{R}^{N\times k}$, $\tilde{A}$ requires $k(M+N)$ instead of $M \cdot N$ units of storage. Additionally, the multiplication of $\tilde{A}$ by a vector x can be done with $\mathcal{O}(k(M+N))$ arithmetic operations instead of $\mathcal{O}(M \cdot N)$. The best rank-k approximation is given by the truncated singular value decomposition; see [12]. The advantage of the latter method over kernel approximation (4) is its black-box nature as it relies only on the entries of A. Since it has cubic complexity, the truncated singular value decomposition cannot be used in practice.

2.1. Partitions and cluster trees

Low-rank approximations are typically employed on suitable blocks and not for the whole matrix. In most cases the approximation of the entire matrix is not possible at all. Therefore, the matrix $A \in \mathbb{R}^{M\times N}$ is decomposed into blocks $t \times s$, $t \subset I := \{1, \dots, M\}$ and $s \subset J := \{1, \dots, N\}$ at first. After that each suitable block A_{ts} is approximated with a low-rank matrix

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

where the number k is small compared to $|t|$ and $|s|$. Let $\operatorname{supp}\Lambda_1 = X_t$ and $\operatorname{supp}\Lambda_2 = X_s$ be the part of the geometry that corresponds to the index sets t and s, respectively. For example, in the case of Galerkin discretizations X_t denotes the union of the supports $X_i := \operatorname{supp}\varphi_i$, $i \in t$.

A block is suitable or *admissible* for approximation if it satisfies the condition

$$\min\{\operatorname{diam} X_t, \operatorname{diam} X_s\} < \beta \operatorname{dist}(X_t, X_s) \tag{5}$$

for a given $\beta > 0$. The expression

$$\operatorname{diam} X = \sup_{x,y\in X} |x-y| \quad \text{and} \quad \operatorname{dist}(X,Y) = \inf_{x\in X, y\in Y} |x-y|$$

are the diameter and the distance of two bounded sets $X, Y \subset \Omega$. Condition (5) guarantees the existence of low-rank approximations if A discretizes an integral representation or the inverse of second-order elliptic partial differential operators; see [5].

Given two meshes represented by the index sets I and J, a partition P of the matrix indices $I \times J$ consisting of admissible blocks or blocks which are small can be found as the leaves of a block-cluster tree $T_{I\times J}$; see [18,5]. This quad-tree can be constructed from two separate binary cluster trees T_I and T_J with roots I and J, respectively. The sons $S_I(t) = \{t', t''\} \subset T_I$ of each node $t \in T_I$ (or $s \in T_J$), if they exist, satisfy $t' \cup t'' = t$ and $t' \cap t'' = \emptyset$. The leaves of T_I are gathered in the set $\mathcal{L}(T_I) := \{t \in T_I : S_I(t) = \emptyset\}$. Applying the mapping S_I recursively, a cluster tree T_I can be constructed consisting of several levels $T_I^{(l)}$, $l = 0, \dots, L$, where L denotes the depth of the tree. Once both cluster trees T_I and T_J have been generated, the block-cluster tree $T_{I\times J}$ can be constructed by recursively subdividing $I \times J$ by following the trees T_I for the rows and T_J for the columns until either (5) is satisfied or the

clusters cannot be subdivided further. As a result, the partition P consists of admissible blocks P_{adm} and non-admissible blocks $P_{\text{non-adm}}$, i.e.

$$P := \mathcal{L}(T_{I\times J}) = P_{\text{adm}} \cup P_{\text{non-adm}}.$$

Note that this does not represent a refinement of the mesh. P is just a partition of the matrix index set $I \times J$ representing the combination of two given meshes. The *sparsity constant* c_{sp} (see [15]) is defined as

$$c_{\text{sp}} := \max\left\{ \max_{t\in T_I} c_{\text{sp,r}}(t), \max_{s\in T_J} c_{\text{sp,c}}(s) \right\},$$

where

$$c_{\text{sp,r}}(t) := |\{s \subset J : t\times s \in P\}|,$$

denotes the maximum number of blocks $t \times s$ contained in P for a given cluster $t \in T_I$ and

$$c_{\text{sp,c}}(s) := |\{t \subset I : t\times s \in P\}|$$

the maximum number of blocks $t \times s \in P$ for a cluster $s \in T_J$. We refer the reader to [5] for more details on the construction of cluster trees.

2.2. Hierarchical matrices and adaptive cross approximation

In view of the construction of the partition P, the set of $\mathcal{H}$-matrices with blockwise rank at most k is defined by

$$\mathcal{H}(P,k) := \{M \in \mathbb{R}^{I\times J} : \text{rank } M_b \le k \text{ for all } b \in P\},$$

see [17,18]. A great advantage of hierarchical matrices is the efficient matrix-vector multiplication. The product of an $\mathcal{H}$-matrix with a vector can be computed in logarithmic-linear time; see [17,18,5].

Many different methods have been developed to generate low-rank approximations on admissible matrix blocks. Replacing the kernel function of the integral operator by truncated kernel expansions as it is described in the beginning of Sect. 4 of this article is a common analytical approach. Examples for such expansions are the multipole expansion [22,16] or interpolating polynomials [8,9]. Other approaches such as the algebraic pseudo-skeleton method [14] work directly on the entries of the considered block. In this article we rely on the adaptive cross approximation (ACA) (see [4]) which requires only few of the original entries to construct the low-rank approximation. Non-admissible blocks cannot be approximated. However, they are small and can be computed entry by entry.

In the following we concentrate on a single admissible block $A_{ts} \in \mathbb{R}^{t\times s}$ of A, where $t \subset I := \{1,\dots,M\}$ and $s \subset J := \{1,\dots,N\}$. According to this notation $A_{i_k,s}$ denotes the i_k-th row, A_{t,j_k} the j_k-th column and A_{i_k,j_k} the entry i_k, j_k. The following Algorithm 1 (see [4,7]) constructs two sequences $\{u_k\} \subset \mathbb{R}^t$ and $\{v_k\} \subset \mathbb{R}^s$. The matrix

$$S_k := \sum_{l=1}^{k} u_l v_l^T$$

has rank equal at most k. Given $\varepsilon_{\text{ACA}} > 0$, the remainder $R_k := A_{ts} - S_k$ has relative accuracy

$$\|R_k\|_F \le \varepsilon_{\text{ACA}} \|A_{ts}\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, such that S_k can be used as an approximation of A_{ts}.

It is easily seen that the vectors u_k and v_k have the representation

$$u_k = (R_{k-1})_{t,j_k} \quad \text{and} \quad v_k = \frac{1}{(R_{k-1})_{i_k,j_k}} (R_{k-1})_{i_k,s}.$$

Remark 1. In the analysis of the error of the adaptive cross approximation, the remainder R_k is associated with the best approximation error of a suitable system $\{\xi_1,\dots,\xi_k\}$ of functions. When selecting the row indices i_k, it must be ensured that the Vandermonde matrix $[\xi_j(x_i)]_{ij} \in \mathbb{R}^{k\times k}$ corresponding to the system in which the approximation error is to be estimated is not singular; cf. [5]. In [5], also several rules are given on how to choose a pivot i_k.

In the case of kernel functions of the form $\kappa(x,y) = \sigma(x)\zeta(y)|x-y|^{-\alpha}$ with $\alpha > 0$ and σ and ζ depending on only one of the variables x and y, respectively, no attention has to be paid to the choice of the row indices, because in this case a system of functions can be specified which leads to a non-singular Vandermonde matrix with a bounded smallest eigenvalue; see [3].

The vanishing rows of the remainders R_k are gathered in the set Z. If the i_k-th row of R_k is nonzero and therefore used as v_k, it is also included in Z as the i_k-th row of the next remainder R_{k+1} vanishes. The number of elements of Z usually depends

Algorithm 1 Adaptive Cross Approximation (ACA).

Input: $A_{ts} \in \mathbb{R}^{t\times s}$, $\varepsilon_{\mathrm{ACA}} > 0$
Output: two sequences $u_k \subset \mathbb{R}^t$ and $v_k \subset \mathbb{R}^s$
Let $k = 1$; $Z = \emptyset$
repeat
find i_k by an appropriate rule (more details in Remark 1)
$\tilde{v}_k := A_{i_k,s}$
for $l = 1, \dots, k-1$ **do** $\tilde{v}_k := \tilde{v}_k - (u_l)_{i_k} v_l$
end for
$Z := Z \cup \{i_k\}$
if $\tilde{v}_k$ does not vanish **then**
$j_k := \operatorname{argmax}_{j\in s} |(\tilde{v}_k)_j|$; $v_k := (\tilde{v}_k)_{j_k}^{-1} \tilde{v}_k$
$u_k := A_{t,j_k}$
for $l = 1, \dots, k-1$ **do** $u_k := u_k - (v_l)_{j_k} u_l$
end for
$k := k + 1$
end if
until $\|u_{k+1}\|_2 \|v_{k+1}\|_2 \le \frac{\varepsilon_{\mathrm{ACA}}(1-\beta)}{1+\varepsilon_{\mathrm{ACA}}} \|S_k\|_F$ or $Z = t$

logarithmically on the desired blockwise precision $\varepsilon_{\mathrm{ACA}}$; see [5]. First, $|Z|$ is to be managed for each block in the case of the matrix-vector multiplication by an adaptive algorithm, where the quality of the approximation of the respective block of A is adapted to the structure of the vector x to be multiplied rather than to the blockwise accuracy $\varepsilon_{\mathrm{ACA}}$.

3. The adaptive matrix-vector multiplication

The goal of this section is to introduce an approximate and adaptive algorithm for the multiplication of a matrix $A \in \mathbb{R}^{M\times N}$ by a vector $x \in \mathbb{R}^N$, i.e.

$$b = Ax,$$

where A is the discretization of a non-local operator and b denotes the resulting vector. Since A is fully populated, the usual way of treating such problems in our case is to approximate the system matrix by hierarchical matrices at first and then to multiply the approximation of A by the vector x. As a result of the construction of the approximation by ACA, redundant and unnecessary information can arise for the simple reason that ACA treats each matrix block independently such that a prescribed accuracy is guaranteed. In order to avoid the generation of such information, we follow an adaptive strategy. Instead of the previous approach of generating a single hierarchical matrix approximation of A, we construct a sequence of approximations A_k and the resulting vectors $b_k := A_k x$. The individual approximations are steered using a residual error estimator based on the h–$h/2$ strategy [13] and the Dörfler marking technique [11]. Note that in contrast to the conventional field of application of such error estimators, no refinement of the geometry or the grid is considered here. While the low-rank approximations of the individual blocks are successively improved, the underlying grid structure and the underlying block-cluster tree are not changed at any time.

Of course the above procedure looks much more complex than multiplying a single approximation of A by the vector x. The approach here aims to exploit properties of the vector x in combination with properties of A. As an example, consider the case that $x = e_i$, $i \in \{1, ..., N\}$, is one of the canonical unit vectors of $\mathbb{R}^N$. Then, the adaptive approach detects that there is no use in computing an approximation of A with accuracy $\varepsilon_{\mathrm{ACA}}$ for blocks not containing parts of column i, while the usual approach would first approximate every block with this accuracy and then perform the multiplication. Depending on the combination of A and x, we expect improved memory requirements and computational time.

A reliable estimate of the error requires the existence of a more accurate approximation $\hat{A}_k$ of A than A_k, i.e., we assume that the saturation assumption

$$\|\hat{b}_k - b\|_2 \le c_{\mathrm{sat}} \|b_k - b\|_2 \tag{6}$$

for some $0 < c_{\mathrm{sat}} < 1$ is fulfilled, where $\hat{b}_k := \hat{A}_k x$. This kind of assumption is quite common in the area of adaptive finite element methods. Notice that in our setting a strategy for approaching (6) is to perform sufficiently many additional ACA steps, which can be observed in the numerical examples for the block-adaptive ACA for linear elasticity. This way of dealing with the saturation assumption is motivated by the exponential convergence of ACA, see [5].

A natural choice (so-called look-ahead approximation) for $\hat{A}_k$ is the improved approximation that results from A_k by applying a fixed number of additional ACA steps to each admissible block and by setting $(\hat{A}_k)_{ts} = A_{ts}$ for all other non-admissible blocks $t \times s \in P_{\text{non-adm}}$. Using the error estimator

$$\gamma_k := \|b_k - \hat{b}_k\|_2 = \| \sum_{t\times s \in P} (A_k - \hat{A}_k)_{ts} x_s \|_2,$$

which is localized with respect to blocks in P, the algorithm for the adaptive matrix-vector multiplication is summarized in Algorithm 2. Following the ideas above, we thus combine the assembly of the discretized non-local operator with the simultaneous computation of the matrix-vector multiplication. Fig. 1 shows a schematic illustration of the procedure.

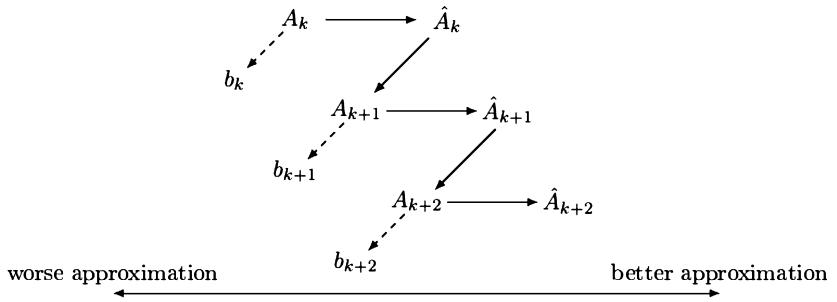

Fig. 1. Schematic illustration of the procedure.

Algorithm 2 Adaptive matrix-vector multiplication (AMVM).

Input: System matrix $A \in \mathbb{R}^{M\times N}$, vector $x \in \mathbb{R}^N$, partition P, $0 < \theta < 1$, $\varepsilon_{\text{AMVM}} > 0$
Output: approximation $b_k = A_k x$ of $b = Ax$

1. Start with a coarse $\mathcal{H}$-matrix approximation A_0 of A and set $k = 0$.
2. Compute $b_k = A_k x$ and $\hat{b}_k = \hat{A}_k x$, where $\hat{A}_k$ is a more accurate approximation of A than A_k, i.e., we assume that the saturation assumption (6) holds.
3. a) Given $0 < \theta < 1$, find a set of marked blocks $P_k \subset P$ with minimal cardinality such that

$$\gamma_k - \gamma_k(P_k) \geq \theta\gamma_k, \tag{7}$$

where $\gamma_k(Q) := \|\sum_{t\times s\in P\setminus Q}(A_k - \hat{A}_k)_{ts}x_s\|_2$ and $\gamma_k = \gamma_k(\emptyset) = \|b_k - \hat{b}_k\|_2$.

b) Use the following strategy to construct P_k:
   (i) Sort the errors $|(b_k - \hat{b}_k)_i|$, $i = 1, \dots, M$, in decreasing order.
   (ii) Go through the ordered errors step by step starting from the top and detect the corresponding blocks in the considered row i.
   (iii) Add every block $t \times s$ to P_k for which $|[(A_k - \hat{A}_k)_{ts}x_s]_i| \geq (1-\theta)(c_{\text{sp}}MNL)^{-1/2}\gamma_k$ holds.
   (iv) Extend P_k according to (ii) and (iii) as long as condition (7) is not fulfilled.

4. Let

$$A_{k+1} = \begin{cases} (\hat{A}_k)_b, & b \in P_k, \\ (A_k)_b, & b \in P \setminus P_k. \end{cases}$$

5. If $\gamma_{k+1} > \varepsilon_{\text{AMVM}}$ increment k and go to 2.

At first glance Algorithm 2 uses two $\mathcal{H}$-matrices A_k and $\hat{A}_k$. Since they are strongly related to each other, it is actually sufficient to store only the more accurate approximation $\hat{A}_k$. Due to the selection criteria of P_k in Algorithm 2, clusters of zero entries in the vector x have the consequence that the associated blocks do not have to be approximated at all. Hence, this approach allows to take into account the structure of the vector x when approximating A. In order to do this we will have to take the depth L of the cluster tree into consideration.

Remark 2. Notice that the Algorithm 2 terminates either if in step 3 b) (iv) condition (7) is satisfied or if the list of blocks has come to its end. In this case also (7) is valid, because the condition used in step 3 b) iii) implies $|[(A_k - \hat{A}_k)_{ts}x_s]_i| \leq (1-\theta)(c_{\text{sp}}MNL)^{-1/2}\gamma_k$ for all blocks $t \times s \in P \setminus P_k$ and thus

$$\begin{aligned}\gamma_k(P_k) &= \|\sum_{t\times s\in P\setminus P_k}(A_k - \hat{A}_k)_{ts}x_s\|_2 = \left(\sum_{i=1}^{M}\left|\left[\sum_{t\times s\in P\setminus P_k}(A_k - \hat{A}_k)_{ts}x_s\right]_i\right|^2\right)^{1/2} \\ &\leq \left(\sum_{i=1}^{M} c_{\text{sp}}L\sum_{t\times s\in P\setminus P_k}|\left[(A_k - \hat{A}_k)_{ts}x_s\right]_i|^2\right)^{1/2} \\ &\leq \left(\sum_{i=1}^{M}\sum_{t\times s\in P\setminus P_k, i\in t}(1-\theta)^2(MN)^{-1}\gamma_k^2\right)^{1/2} \leq (1-\theta)\gamma_k.\end{aligned}$$

The newly introduced algorithm will be examined in more detail in the next steps. First, we consider the reliability and the efficiency as two basic characteristics of the error estimator.

Lemma 1. *Let assumption (6) be valid. Then γ_k is efficient and reliable, i.e., it holds*

$$c_{\text{eff}}\gamma_k \leq \|b_k - b\|_2 \leq c_{\text{rel}}\gamma_k,$$

where $c_{\text{eff}} := 1/(1 + c_{\text{sat}})$ and $c_{\text{rel}} := 1/(1 - c_{\text{sat}})$.

Proof. With the saturation assumption it follows

$$\|b_k - b\|_2 \le \|b_k - \hat{b}_k\|_2 + \|\hat{b}_k - b\|_2 \le \gamma_k + c_{\text{sat}}\|b_k - b\|_2$$

and thus

$$\|b_k - b\|_2 \le c_{\text{rel}}\gamma_k,$$

which proves the reliability of the estimator γ_k. Using again the saturation assumption, we obtain

$$\gamma_k = \|b_k - \hat{b}_k\|_2 \le \|b_k - b\|_2 + \|b - \hat{b}_k\|_2 \le (1 + c_{\text{sat}})\|b_k - b\|_2$$

and thus

$$c_{\text{eff}}\gamma_k \le \|b_k - b\|_2. \quad \square$$

The next property of the estimator γ_k which has to be investigated is the estimator convergence. In order to do this, we must first examine the behavior of the error $\hat{e}_k := \|\hat{b}_k - \hat{b}_{k+1}\|_2$, where $\hat{b}_k = \hat{A}_k x$.

Lemma 2. *The error $\hat{e}_k$ converges to zero for $k \to \infty$.*

Proof. For $\hat{e}_k$ we observe

$$\hat{e}_k^2 = \|\hat{A}_k x - \hat{A}_{k+1}x\|_2^2 \le \|\hat{A}_k - \hat{A}_{k+1}\|_2^2\|x\|_2^2.$$

Notice that each block $t \times s \in P$ is either not chosen in Algorithm 2 from some index $k_0 \in \mathbb{N}$ on, i.e. $(A_k)_{ts} = (A_{k_0})_{ts}$ for $k \ge k_0$ or $\lim_{k\to\infty}(A_k)_{ts} = A_{ts}$, which follows from the fact that ACA reproduces the original matrix after at most $\min\{|t|, |s|\}$ steps, see [5]. Hence, we have

$$\lim_{k\to\infty} A_k = A_*$$

with some matrix $A_* \in \mathbb{R}^{N\times N}$ and thus $\lim_{k\to\infty} \|\hat{A}_{k+1} - \hat{A}_k\| = 0$. $\square$

The convergence of the error estimator γ_k can be proven via an estimator reduction principle, which was originally introduced in [1] in the context of the adaptive boundary element method.

Lemma 3 *(Estimator reduction). Let $s > 1$ and $1 - \frac{1}{\sqrt{s}} < \theta < 1$ be given. Then it holds that*

$$\gamma_{k+1}^2 \le c_1\gamma_k^2 + c_2\hat{e}_k^2,$$

where $c_1 = 1/s < 1$ and $c_2 = [1 - s(1-\theta)^2]^{-1}$. Furthermore, $\lim_{k\to\infty}\gamma_k = 0$.

Proof. We have a closer look at the error estimator γ_{k+1}. With $\delta > 0$ and Young's inequality it follows

$$\begin{aligned}\gamma_{k+1}^2 &= \|b_{k+1} - \hat{b}_{k+1}\|_2^2 = \|b_{k+1} - \hat{b}_k + \hat{b}_k - \hat{b}_{k+1}\|_2^2 \\ &\le \left(\|b_{k+1} - \hat{b}_k\|_2 + \|\hat{b}_k - \hat{b}_{k+1}\|_2\right)^2 \\ &\le (1+\delta)\underbrace{\|b_{k+1} - \hat{b}_k\|_2^2}_{=:e^2} + (1 + 1/\delta)\hat{e}_k^2.\end{aligned}$$

If we split up e into the marked and non-marked blocks, from (7) we obtain the following estimator reduction:

$$\begin{aligned}e = \|b_{k+1} - \hat{b}_k\|_2 &= \|\sum_{t\times s\in P}(A_{k+1} - \hat{A}_k)_{ts}x_s\|_2 \\ &= \|\sum_{t\times s\in P_k}(A_{k+1} - \hat{A}_k)_{ts}x_s + \sum_{t\times s\in P\setminus P_k}(A_{k+1} - \hat{A}_k)_{ts}x_s\|_2 \\ &= \|\sum_{t\times s\in P\setminus P_k}(A_k - \hat{A}_k)_{ts}x_s\|_2 = \gamma_k(P_k) \le (1-\theta)\gamma_k.\end{aligned}$$

With the choice $\delta = \frac{1-s(1-\theta)^2}{s(1-\theta)^2}$ we get

$$\gamma_{k+1}^2 \le (1+\delta)(1-\theta)^2\gamma_k^2 + (1 + 1/\delta)\hat{e}_k^2$$

$$= \left(1 + \frac{1 - s(1-\theta)^2}{s(1-\theta)^2}\right)(1-\theta)^2\gamma_k^2 + \left(1 + \frac{s(1-\theta)^2}{1 - s(1-\theta)^2}\right)\hat{e}_k^2$$
$$= \frac{1}{s}\gamma_k^2 + \frac{1}{1 - s(1-\theta)^2}\hat{e}_k^2.$$

The second part of the assertion is proved with Lemma 2 and the error estimator reduction principle introduced in [1]. Let $\hat{E} > 0$ be a number satisfying $\hat{e}_k \le \hat{E}$ for all k. The estimator reduction principle leads to

$$\begin{aligned}\gamma_{k+1}^2 &\le c_1\gamma_k^2 + c_2\hat{e}_k^2 \le c_1(c_1\gamma_{k-1}^2 + c_2\hat{e}_{k-1}^2) + c_2\hat{e}_k^2 \\ &= c_1^2\gamma_{k-1}^2 + c_1c_2\hat{e}_{k-1}^2 + c_2\hat{e}_k^2 \\ &\le \dots \le c_1^{k+1}\gamma_0^2 + c_2\sum_{i=0}^{k} c_1^{k-i}\hat{e}_k^2 \\ &\le c_1^{k+1}\gamma_0^2 + c_2\hat{E}\sum_{l=0}^{k} c_1^l \le \gamma_0^2 + \frac{c_2\hat{E}}{1-c_1}.\end{aligned}$$

Accordingly, the sequence $\{\gamma_k\}_{k\in\mathbb{N}_0}$ is bounded and we are able to define $\Gamma := \limsup_{k\to\infty}\gamma_k^2$. Using the estimator reduction principle once more yields

$$\Gamma = \limsup_{k\to\infty}\gamma_{k+1}^2 \le c_1\limsup_{k\to\infty}\gamma_k^2 + c_2\underbrace{\limsup_{k\to\infty}\hat{e}_k^2}_{=0} = c_1\Gamma.$$

Thus $\Gamma = 0$ and it follows

$$0 \le \liminf_{k\to\infty}\gamma_k \le \limsup_{k\to\infty}\gamma_k = \Gamma = 0,$$

which shows

$$\lim_{k\to\infty}\gamma_k = 0. \quad \square$$

Exploiting the reliability of the error estimator, the convergence of the adaptive matrix-vector multiplication can also be shown.

Lemma 4 *(Estimator convergence). Let the requirements of Lemma 1 and Lemma 3 be valid. Then, the error $\|b_k - b\|_2$ of the sequence $\{b_k\}_{k\in\mathbb{N}}$ constructed by Algorithm 2 converges to zero.*

Proof. Using the reliability of the estimator and the reduction principle leads to

$$\|b_k - b\|_2 \le c_{\mathrm{rel}}\gamma_k \to 0 \quad \text{for } k \to \infty. \quad \square$$

4. Boundary integral approximation of linear elasticity

4.1. Integral formulation of linear elasticity

We assume that $\Omega \subset \mathbb{R}^3$ is a Lipschitz domain and its boundary $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, is partitioned into a Dirichlet boundary Γ_D and a Neumann boundary Γ_N.

The solution of the equations of linear elasticity, see for instance [24], can be written as

$$u(x) = \tilde{V}\gamma_1^{\mathrm{int}}u - \tilde{W}\gamma_0^{\mathrm{int}}u,$$

where

$$\tilde{V}f(x) := \int_{\partial\Omega} S(x,y)f(y)\,\mathrm{d}s_y, \quad f \in [H^{-1/2}(\partial\Omega)]^3,$$

denotes the single-layer potential and

$$\tilde{W}g(x) := \int_{\partial\Omega} \gamma_{1,y}^{\mathrm{int}}S(x,y)g(y)\,\mathrm{d}s_y, \quad g \in [H^{1/2}(\partial\Omega)]^3,$$

denotes the double-layer potential and $\gamma_1^{\mathrm{int}}u$ the conormal derivative. In the case of linear elasticity, the conormal derivative takes the form

$$\gamma_1^{\text{int}} u = \lambda \operatorname{div} u \cdot n + 2\mu\, n \cdot \nabla u + \mu\, n \times \operatorname{curl} u$$

with the normal vector n. The fundamental solution $S(x,y) := S_K(x-y)$ of linear elasticity is given by Kelvin's solution tensor

$$S_K(x) := \left(\frac{1}{8\pi} \frac{1}{E} \frac{1+\nu}{1-\nu} \left[\frac{3-4\nu}{|x|} \delta_{ij} + \frac{x_i x_j}{|x|^3} \right] \right)_{ij} \in \mathbb{R}^{3\times 3}$$

for $x \in \mathbb{R}^3 \setminus \{0\}$. Using the trace operators γ_0^{int} and γ_1^{int}, we define the single-layer operator $V := \gamma_0^{\text{int}} \tilde{V}$ and the hyper-singular operator $D := -\gamma_1^{\text{int}} \tilde{W}$. Furthermore, we are going to use the double-layer operator

$$(Kg)(x) = \lim_{\varepsilon \to 0} \int_{\partial\Omega \setminus B_\varepsilon(x)} \gamma_{1,y}^{\text{int}} S(x,y) g(y)\, \mathrm{d}s_y, \quad x \in \partial\Omega, \quad g \in [H^{1/2}(\partial\Omega)]^3$$

and its adjoint

$$(K'f)(x) = \lim_{\varepsilon \to 0} \int_{\partial\Omega \setminus B_\varepsilon(x)} \gamma_{1,x}^{\text{int}} S(x,y) f(y)\, \mathrm{d}s_y, \quad x \in \partial\Omega, \quad f \in [H^{-1/2}(\partial\Omega)]^3.$$

In order to solve mixed boundary value problems (cf. Sect. 1), boundary integral operators have to be defined on the respective part of the boundary. On the Dirichlet boundary Γ_D we set

$$V_{DD} : [\tilde{H}^{-1/2}(\Gamma_D)]^3 \to [H^{1/2}(\Gamma_D)]^3, \quad V_{DD} f := (V\tilde{f})|_{\Gamma_D},$$

where $\tilde{H}^{-1/2}(\Gamma_D) = [H^{1/2}(\Gamma_D)]'$ and $f = \tilde{f}|_{\Gamma_D}$ with $\tilde{f} \in [H^{-1/2}(\partial\Omega)]^3$ and $\operatorname{supp} \tilde{f} \subset \Gamma_D$. Using the extension $\tilde{g} \in [H^{1/2}(\partial\Omega)]^3$ of a function $g \in [\tilde{H}^{1/2}(\Gamma_N)]^3$, where

$$\tilde{H}^{1/2}(\Gamma_N) := \{ v = \tilde{v}|_{\Gamma_N} : \tilde{v} \in H^{1/2}(\partial\Omega), \operatorname{supp} \tilde{v} \subset \Gamma_N \},$$

we define

$$D_{NN} : [\tilde{H}^{1/2}(\Gamma_N)]^3 \to [H^{-1/2}(\Gamma_N)]^3, \quad D_{NN} g := (D\tilde{g})|_{\Gamma_N}$$

with $H^{-1/2}(\Gamma_N) = [\tilde{H}^{1/2}(\Gamma_N)]'$. The following two operators describe the interaction between the Dirichlet and the Neumann data. We define the double-layer operator of the Neumann boundary

$$K_{ND} : [\tilde{H}^{1/2}(\Gamma_N)]^3 \to [H^{1/2}(\Gamma_D)]^3, \quad K_{ND} g := (K\tilde{g})|_{\Gamma_D}$$

and the adjoint double-layer operator of the Dirichlet boundary

$$K'_{DN} : [\tilde{H}^{-1/2}(\Gamma_D)]^3 \to [H^{-1/2}(\Gamma_N)]^3, \quad K'_{DN} f := (K'\tilde{f})|_{\Gamma_N}.$$

Then, the boundary value problem (1) with the boundary conditions (2) and (3) has the solution

$$v = \tilde{V}(\tilde{g}_N + \tilde{t}) - \tilde{W}(\tilde{g}_D + \tilde{u}), \tag{8}$$

where $\tilde{u} \in [H^{1/2}(\partial\Omega)]^3$ and $\tilde{t} \in [H^{-1/2}(\partial\Omega)]^3$ denote the extensions by zero of the functions $u \in [\tilde{H}^{1/2}(\Gamma_N)]^3$ and $t \in [\tilde{H}^{-1/2}(\Gamma_D)]^3$, which are the solutions of the integral equations

$$V_{DD} t - K_{ND} u = \left(\frac{1}{2} I + K \right) \tilde{g}_D|_{\Gamma_D} - V\tilde{g}_N|_{\Gamma_D},$$
$$K'_{DN} t + D_{NN} u = -D\tilde{g}_D|_{\Gamma_N} + \left(\frac{1}{2} I - K' \right) \tilde{g}_N|_{\Gamma_N}.$$

In (8) the given Dirichlet data $g_D \in [H^{1/2}(\Gamma_D)]^3$ and Neumann data $g_N \in [H^{-1/2}(\Gamma_N)]^3$ are extended to the functions $\tilde{g}_D \in [H^{1/2}(\partial\Omega)]^3$ and $\tilde{g}_N \in [H^{-1/2}(\partial\Omega)]^3$; see [24].

A stable numerical treatment of the above operators is only possible if the singularities are not too strong. Since V is weakly singular, we do not expect any numerical problems. For D and K weakly singular representations have to be found. Using the boundary differential operators

$$T_{ij}(x) := n_j(x) \frac{\partial}{\partial x_j} - n_i(x) \frac{\partial}{\partial x_i}, \quad i,j = 1,2,3, \quad x \in \partial\Omega,$$

and

$$\frac{\partial}{\partial S_1}(x) := T_{32}(x), \quad \frac{\partial}{\partial S_2}(x) := T_{13}(x), \quad \frac{\partial}{\partial S_3}(x) := T_{12}(x),$$

the double-layer operator K can be rewritten as

$$K\tilde{u}(x) = \frac{1}{4\pi}\int_{\partial\Omega}\frac{\partial}{\partial n_y}\frac{1}{|x-y|}\tilde{u}(y)\,\mathrm{d}s_y - \frac{1}{4\pi}\int_{\partial\Omega}\frac{1}{|x-y|}T\tilde{u}(y)\,\mathrm{d}s_y + 2\mu VT\tilde{u}(x)$$

for $\tilde{u}\in[H^{1/2}(\partial\Omega)]^3$. The operator D in terms of weakly singular integrals has the representation

$$\begin{aligned}\langle D\tilde{u},\tilde{v}\rangle_{\partial\Omega} &= \frac{\mu}{4\pi}\int_{\partial\Omega}\int_{\partial\Omega}\frac{1}{|x-y|}\left(\sum_{k=1}^{3}\frac{\partial}{\partial S_k}\tilde{u}(y)\cdot\frac{\partial}{\partial S_k}\tilde{v}(x)\right)\mathrm{d}s_x\,\mathrm{d}s_y\\ &\quad+\frac{\mu}{2\pi}\int_{\partial\Omega}\int_{\partial\Omega}(T(x)\tilde{v}(x))^T\frac{I}{|x-y|}(T(y)\tilde{u}(y))^T\,\mathrm{d}s_x\,\mathrm{d}s_y - 4\mu^2\langle VT\tilde{u},T\tilde{v}\rangle_{\partial\Omega}\\ &\quad+\frac{\mu}{4\pi}\int_{\partial\Omega}\int_{\partial\Omega}\sum_{i,j,k=1}^{3}T_{kj}(x)\tilde{v}_i(x)\frac{1}{|x-y|}T_{ki}(y))\tilde{v}_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y;\end{aligned}$$

see [19].

4.2. Discretization techniques

Our goal is the computation of a numerical solution of the integral equations for linear elasticity via the boundary element method (BEM). The starting point is an admissible triangulation $\mathcal{T}_h$ of the surface of the computational domain Ω in regular triangles τ_i, $i=1,\dots,M$, and nodes p_j, $j=1,\dots,N$. Here a triangulation is called admissible, if neighboring triangles have only one common edge or node; see [23].

On the triangulation $\mathcal{T}_h$ the space of piecewise constant functions S_h^0 is given by its basis

$$\varphi_i(x) = \begin{cases}1, & x\in\tau_i,\\ 0, & \text{else},\end{cases}\qquad i=1,\dots,M,$$

which is used for the discretization of the operator V. The operator K and parts of D are discretized using functions

$$\psi_j(x) := \begin{cases}1, & x=p_j,\\ 0, & x=p_l \text{ for } l\in\{1,\dots,N\}\setminus\{j\},\\ \text{linear}, & \text{else},\end{cases}\qquad j=1,\dots,N,$$

defining a basis of the space $S_h^1(\Gamma)$ of continuous and piecewise linear functions. We find solutions of the form

$$\tilde{t}_h(x) = \sum_{i=1}^{M}\begin{bmatrix}\tilde{t}_i^{(1)}\\ \tilde{t}_i^{(2)}\\ \tilde{t}_i^{(3)}\end{bmatrix}\varphi_i(x)\quad\text{and}\quad \tilde{u}_h(x) = \sum_{j=1}^{N}\begin{bmatrix}\tilde{u}_j^{(1)}\\ \tilde{u}_j^{(2)}\\ \tilde{u}_j^{(3)}\end{bmatrix}\psi_j(x).$$

The coefficient vectors $\tilde{t}=[\tilde{t}_i^{(1)},\tilde{t}_i^{(2)},\tilde{t}_i^{(3)}]_{i=1}^M$ and $\tilde{u}=[\tilde{u}_j^{(1)},\tilde{u}_j^{(2)},\tilde{u}_j^{(3)}]_{j=1}^N$ are the solution of the linear system of equations

$$\begin{bmatrix}V_{DD,h} & -K_{ND,h}\\ K_{ND,h}^T & D_{NN,h}\end{bmatrix}\begin{bmatrix}\tilde{t}\\ \tilde{u}\end{bmatrix} = \begin{bmatrix}-V & \frac{1}{2}M+K\\ \frac{1}{2}M^T-K^T & -D\end{bmatrix}\begin{bmatrix}\tilde{g}_N\\ \tilde{g}_D\end{bmatrix} =: \begin{bmatrix}f_D\\ f_N\end{bmatrix} \tag{9}$$

$$V_{DD,h}[ij] = \langle V_{DD}\varphi_j,\varphi_i\rangle_{\Gamma_D},\quad K_{ND,h}[jk] = \langle K_{ND}\psi_k,\varphi_j\rangle_{\Gamma_D},$$

$$D_{NN,h}[kl] = \langle D_{NN}\psi_l,\psi_k\rangle_{\Gamma_N}$$

for $i,j=1,\dots,M$ and $k,l=1,\dots,N$.

At the end of this section we want to state representations for the operators $V_{DD,h}$, $K_{ND,h}$, and $D_{NN,h}$, which are more advantageous for numerical calculations. Using Kelvin's solution tensor and a suitable space for the discretization of the operators like the space $[S^0(\Gamma)]^3$, the stiffness matrix $V_h\in\mathbb{R}^{3M\times 3M}$ of the single-layer potential has the representation

$$V_h = \frac{1}{2}\frac{1}{E}\frac{1+\nu}{1-\nu}\left((3-4\nu)\begin{bmatrix}V_{\Delta,h} & 0 & 0\\ 0 & V_{\Delta,h} & 0\\ 0 & 0 & V_{\Delta,h}\end{bmatrix}+\begin{bmatrix}V_{11} & V_{12} & V_{13}\\ V_{12} & V_{22} & V_{23}\\ V_{13} & V_{23} & V_{33}\end{bmatrix}\right), \tag{10}$$

where

$$V_{\Delta,h}[ij] = \frac{1}{4}\int_{\tau_j}\int_{\tau_i}\frac{1}{|x-y|}\,\mathrm{d}s_y\,\mathrm{d}s_x$$

and

$$V_{kl}[ij] = \frac{1}{4}\int_{\tau_j}\int_{\tau_i} \frac{(x_k - y_k)(x_l - y_l)}{|x-y|^3}\,\mathrm{d}s_y\,\mathrm{d}s_x$$

are $M \times M$ sub-matrices for $k,l = 1,2,3$. Together with the space $[S^1(\Gamma)]^3$, the operator K_h can be represented as

$$K_h = \begin{bmatrix} K_{\Delta,h} & 0 & 0 \\ 0 & K_{\Delta,h} & 0 \\ 0 & 0 & K_{\Delta,h} \end{bmatrix} - \begin{bmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{bmatrix} T_h + \frac{E}{1+\nu} V_h T_h, \tag{11}$$

where

$$K_{\Delta,h}[ij] = \frac{1}{4\pi} \sum_{\tau \in \operatorname{supp}\psi_j} \int_{\tau}\int_{\tau_i} \frac{(x-y)^T n(y)}{|x-y|^3}\psi_j(y)\,\mathrm{d}s_y\,\mathrm{d}s_x,$$

$i = 1,\dots,M$, $j = 1,\dots,N$, and

$$T_h := \begin{bmatrix} 0 & T_{12,h} & T_{13,h} \\ -T_{12,h} & 0 & T_{23,h} \\ -T_{13,h} & -T_{23,h} & 0 \end{bmatrix}, \quad T_{kl,h}[ij] := T_{kl}(\hat{x})\psi_j(\hat{x}),\ \hat{x} \in \tau_i,$$

for $k,l \in \{1,2,3\}$, $i = 1,\dots,M$, $j = 1,\dots,N$. Finally, the matrix D_h is given by

$$\begin{aligned} D_h = \sum_{k=1}^{3} \frac{\mu}{4\pi} S_{k,h}^T \begin{bmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{bmatrix} S_{k,h} + \frac{\mu}{2\pi} T_h^T \begin{bmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{bmatrix} T_h \\ + 4\mu^2 T_h^T V_h T_h + \frac{\mu}{4\pi} D_h' \end{aligned} \tag{12}$$

with

$$D_h' := \begin{bmatrix} D_{11,h}' & D_{12,h}' & D_{13,h}' \\ D_{21,h}' & D_{22,h}' & D_{23,h}' \\ D_{31,h}' & D_{32,h}' & D_{33,h}' \end{bmatrix}, \quad D_{ij,h}' := \sum_{k=1}^{3} T_{kj,h}^T V_{\Delta,h} T_{ki,h}$$

and

$$S_{1,h} := \begin{bmatrix} T_{32,h} & 0 & 0 \\ 0 & T_{32,h} & 0 \\ 0 & 0 & T_{32,h} \end{bmatrix}, \quad S_{2,h} := \begin{bmatrix} T_{13,h} & 0 & 0 \\ 0 & T_{13,h} & 0 \\ 0 & 0 & T_{13,h} \end{bmatrix},$$

$$S_{3,h} := \begin{bmatrix} T_{21,h} & 0 & 0 \\ 0 & T_{21,h} & 0 \\ 0 & 0 & T_{21,h} \end{bmatrix},$$

see [21]. Using specific restriction operators defined in [21] allows the re-presentation of the discretized operators V_h, K_h, and D_h with respect to the corresponding boundaries, resulting in the operators $V_{DD,h}$, $K_{ND,h}$, and $D_{NN,h}$.

Remark 3. Instead of the componentwise representation of the previous discrete operators, a common alternative is a blockwise approach. The advantage of the latter is that the 3×3 matrix blocks can basically be computed at the costs of a single entry. The downside of this approach is that for applying ACA the invertibility of pivot blocks has to be guaranteed. Hence, we decided to use the componentwise approach although the method introduced in this article also can be applied to the blockwise representation.

5. The adaptive solution of Lamé equations

The adaptive matrix-vector multiplication introduced in Sect. 3 can be applied in the context of boundary element methods when the given data vectors are multiplied by discrete integral operators on the right-hand side of the discretized integral equation. If also the solution of the latter is to be computed, then the BACA method introduced in [2] can be employed. It combines the adaptive construction of the $\mathcal{H}$-matrix approximation of the system matrix with the simultaneous iterative solution of the system. The individual blocks are approximated only as accurate as necessary for the prescribed accuracy and the given right-hand side vector. The BACA was developed for the Laplace equation and is thus not adapted to the structure of the Lamé equations. In this section, BACA will be modified to allow its application to problems from linear elasticity.

We consider the numerical solution of the linear system $Ax = b$ from (9) with

$$A = \begin{bmatrix} V_{DD,h} & -K_{ND,h} \\ K_{ND,h}^T & D_{NN,h} \end{bmatrix}, \quad b = \begin{bmatrix} f_D \\ f_N \end{bmatrix}, \quad \text{and} \quad x = \begin{bmatrix} \tilde{t} \\ \tilde{u} \end{bmatrix}.$$

Each of the four sub-matrices of A consists again of nine sub-matrices with an associated block-cluster tree. Let P be the union of all admissible partitions concerning the boundaries and the different operators, i.e.,

$$P := P_V \cup P_K \cup P_D,$$

where P_V, P_K, and P_D consist of all blocks of the discretized single-layer operator V_h, the discretized double-layer operator K_h and the discretized hyper-singular operator D_h and denote by P_{adm} the admissible blocks contained in P. Accordingly, $c_{\text{sp},V}$, $c_{\text{sp},K}$, and $c_{\text{sp},D}$ are the sparsity constants associated with the block-cluster trees for the operators V_h, K_h, and D_h. The constructed matrix approximation is denoted by

$$A_k = \begin{bmatrix} V_{DD,k} & -K_{ND,k} \\ K_{ND,k}^T & D_{NN,k} \end{bmatrix},$$

where the sub-matrices $V_{DD,h}$, $K_{ND,h}$, and $D_{NN,h}$ are approximated individually. Moreover, let

$$\hat{A}_k = \begin{bmatrix} \hat{V}_{DD,k} & -\hat{K}_{ND,k} \\ \hat{K}_{ND,k}^T & \hat{D}_{NN,k} \end{bmatrix}$$

be a more accurate approximation of A than A_k. We assume that the saturation condition

$$\|\hat{A}_k x_k - A x_k\|_2 \le c_{\text{sat}} \|A_k x_k - A x_k\|_2 \tag{13}$$

is fulfilled for some $0 < c_{\text{sat}} < 1$, where x_k denotes the solution of the linear system $A_k x_k = b$. Again, a possible strategy for choosing $\hat{A}_k$ is to add a fixed number of additional ACA steps for each admissible block of A_k (look-ahead approximation) and to set $(\hat{A}_k)_{ts} = A_{ts}$ for all other blocks $t \times s \in P_{\text{non-adm}}$.

In order to obtain some information about the error of the approximation, we use the error estimator

$$\mathcal{E}_k^2 := \sum_{t\times s \in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2.$$

If BACA is to be applied to the saddle-point problem (9), we first have to employ the Bramble-Pasciak conjugate gradient method [10] as the iterative solver. The difference to the case of the Laplace equation lies in the selection of the blocks to be refined. We use the following strategy based on the representations (10), (11), and (12) of the discretized operators V_h, K_h, and D_h, where we leave the fixed matrices T_h, $S_{1,h}$, $S_{2,h}$ and $S_{3,h}$ unchanged during the whole procedure. Since the sub-matrix $V_{\Delta,h}$ is contained in all the operators V_h, K_h, and D_h, the refinement of $V_{\Delta,h}$ is implemented at first. Afterwards the refinements of V_{ij}, $i,j = 1,2,3$, $K_{\Delta,h}$, D_h' and D_h follow. The approximation of a block is improved only if it has been selected. This leads to the following Algorithm 3.

Algorithm 3 Block-adaptive ACA for linear elasticity.

Input: System matrix A, right-hand side b, partition P_{adm}, $0 < \theta < 1$, $\alpha \ge 0$, $\epsilon_{\text{BACA}} > 0$
Output: approximation x_k of the solution x of $Ax = b$

1. Start with a coarse $\mathcal{H}$-matrix approximation A_0 of A and set $k = 0$.
2. Given $\alpha \ge 0$, apply the Bramble-Pasciak-CG to the linear system $A_k x_k = b$ until the residual error satisfies

$$\|b - A_k x_k\|_2 \le \alpha \|(A_k - \hat{A}_k) x_k\|_2 \tag{14}$$

   (use x_{k-1} as a starting vector; $x_{-1} := 0$).
3. Given $0 < \theta < 1$, find a set of marked blocks $M_k \subset P_{\text{adm}}$ with minimal cardinality such that

$$\mathcal{E}_k(M_k) \ge \theta \mathcal{E}_k, \tag{15}$$

   where $\mathcal{E}_k^2(M) := \sum_{t\times s\in M} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2$ and $\mathcal{E}_k := \mathcal{E}_k(P_{\text{adm}})$.
4. Consider the following cases for all $t \times s \in M_k$:
   (i) If $t \times s$ belongs to V_{DD}, set $(V_{\Delta,h,k+1})_{ts} = (\hat{V}_{\Delta,h,k})_{ts}$ and $(V_{ij})_{ts} = (\hat{V}_{ij})_{ts}$, $i,j = 1,2,3$.
   (ii) If $t \times s$ belongs to K_{ND}, set $(K_{\Delta,h,k+1})_{ts} = (\hat{K}_{\Delta,h,k})_{ts}$ and $(V_{\Delta,h,k+1})_b = (\hat{V}_{\Delta,h,k})_b$, $(V_{ij})_b = (\hat{V}_{ij})_b$, $i,j = 1,2,3$, for all blocks b having rows associated with t.
   (iii) If $t \times s$ belongs to D_{NN}, set $V_{\Delta,h,k+1} = \hat{V}_{\Delta,h,k}$ and $V_{ij} = \hat{V}_{ij}$, $i,j = 1,2,3$.
   All blocks not selected remain at the current stage of approximation.
5. If $\mathcal{E}_{k+1} > \epsilon_{\text{BACA}}$ increment k and go to 2.

Due to the structure of the discrete operators (see Section 4.2), the selection strategy in Algorithm 3 is such that the individual blocks do not need to be approximated independently by ACA. For the part V_{DD} the refinement of a block can directly be carried over to the corresponding block in $V_{\Delta,h}$ and V_{ij}. Since the operator D_{NN} contains the operator V_h in the form of a multiplication with the matrices T_h, blocks can no longer be selected directly. In this case, V_h (or at least the restriction of V_h to the Neumann part) must be completely refined.

In the following we adapt the convergence analysis (presented in [2]) to the previous method. For the efficiency of the error estimator or at least a lower bound on the expression $\|b - Ax_k\|_2$, we refer to [2]. The reliability of the estimator follows from the saturation assumption.

Lemma 5. *Let the saturation assumption (13) be valid. Then $\mathcal{E}_k$ is reliable, i.e. it holds*

$$\|b - Ax_k\|_2 \leq \frac{1+\alpha(1+c_{\text{sat}})}{1-c_{\text{sat}}}\|(A_k - \hat{A}_k)x_k\|_2 \leq \sqrt{27C_{\text{sp}}L}\,\frac{1+\alpha(1+c_{\text{sat}})}{1-c_{\text{sat}}}\mathcal{E}_k,$$

where L is the maximum depth of the used cluster trees and $C_{\text{sp}} := \max\{c_{\text{sp}V}, c_{\text{sp},K}, c_{\text{sp},D}\}$.

Proof. The first assertion follows with condition (14) and the saturation assumption from

$$\begin{aligned}\|b - Ax_k\|_2 &\leq \|b - A_k x_k\|_2 + \|(A_k - \hat{A}_k)x_k\|_2 + \|\hat{A}_k x_k - Ax_k\|_2\\ &\leq (\alpha+1)\|(A_k - \hat{A}_k)x_k\|_2 + c_{\text{sat}}\|A_k x_k - Ax_k\|_2\\ &\leq (\alpha + 1 + c_{\text{sat}}\alpha)\|(A_k - \hat{A}_k)x_k\|_2 + c_{\text{sat}}\|b - Ax_k\|_2.\end{aligned}$$

The second inequality is a result of the decomposition of the sub-matrices of $A = \sum_{l=1}^{L} A^{(l)}$ into a sum of level matrices $A^{(l)}$. Due to the fact that the different operators have different cluster trees, a maximum level L will be chosen among these cluster trees. Then, no more further sub-matrices will exist at a certain level. In this case, use the zero sub-matrix for the remaining levels. We observe

$$\begin{aligned}\|(A_k - \hat{A}_k)x_k\|_2^2 &\leq \left(\sum_{l=1}^{L}\|(A_k - \hat{A}_k)^{(l)}x_k\|_2\right)^2 \leq L\sum_{l=1}^{L}\|(A_k - \hat{A}_k)^{(l)}x_k\|_2^2\\ &= L\sum_{l=1}^{L}\sum_{t\in T_I^{(l)}}\Big\|\sum_{s:t\times s\in P}(A_k - \hat{A}_k)_{ts}(x_k)_s\Big\|_2^2\\ &\leq L\sum_{l=1}^{L}\sum_{t\in T_I^{(l)}}\left(\sum_{s:t\times s\in P}\|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2\right)^2\\ &\leq 27C_{\text{sp}}L\sum_{l=1}^{L}\sum_{t\in T_I^{(l)}}\sum_{s:t\times s\in P}\|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2\\ &= 27C_{\text{sp}}L\sum_{t\times s\in P}\|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2\\ &= 27C_{\text{sp}}L\mathcal{E}_k^2,\end{aligned}$$

since each of the three discretized operators consists of nine sub-operators, which explains the factor 27 at the end. □

Except for the inclusion of different sparsity constants and tree depths, there are no other differences in the convergence proof of the adapted BACA method compared to BACA for the Laplace equation. For this reason, we refer to the proofs in [2] for the rest of the convergence analysis and only state the convergence results here.

Lemma 6. *Assume that $A_* := \lim_{k\to\infty} A_k$ is invertible and α is sufficiently small. Then it holds that*

$$\mathcal{E}_{k+1}^2 \leq q\,\mathcal{E}_k^2 + z_k,$$

where z_k converges to zero and $q < 1$. Furthermore, $\lim_{k\to\infty}\mathcal{E}_k = 0$.

Lemma 7. *The residuals $r_k := b - Ax_k$ of the sequence $\{x_k\}_{k\in\mathbb{N}}$ constructed by Algorithm 3 converge to zero.*

At the end of this section, another field of application for AMVM will be briefly discussed. Let $(Y_h, Z_h) = \left(\tilde{g}_{N,h} + \tilde{t}_h, \tilde{g}_{D,h} + \tilde{u}_h\right)$ denote the whole approximated Dirichlet and Neumann data after having calculated the missing data $(\tilde{t}_h, \tilde{u}_h)$, i.e.

$$Y_h = \sum_{i=1}^{M} y_i\varphi_i \quad \text{and} \quad Z_h = \sum_{j=1}^{N} z_j\psi_j$$

with coefficient vectors $y, z \in \mathbb{R}^3$, the solution u_h in Ω can be evaluated by

$$u_h(x) = \sum_{j=1}^{M} y_j \int_{\partial\Omega} S(x,y)\varphi_j(y)\,\mathrm{d}s_y - \sum_{k=1}^{N} z_k \int_{\partial\Omega} \gamma_{1,y}^{\text{int}} S(x,y)\psi_k(y)\,\mathrm{d}s_y,$$

for $x \in \Omega$, and the stresses $\sigma(u_h, x)$ can be computed using the derivatives

Table 1
Error and time required to compute right-hand side of (9) via ACA.

N	M	$b_{\min}$	$\|b - b_{\mathrm{ACA}}\|_2$	time approximation
972	488	15	$3.45 \cdot 10^{-7}$	6.9 s
3 888	1 946	20	$4.43 \cdot 10^{-7}$	39.6 s
15 552	7 778	30	$2.53 \cdot 10^{-7}$	235.7 s
62 208	31 106	40	$2.17 \cdot 10^{-7}$	1428.6 s

$$\partial_{x_i} u_h(x) = \sum_{j=1}^{M} y_j \int_{\partial\Omega} \partial_{x_i} S(x,y)\varphi_j(y)\,\mathrm{d}s_y - \sum_{k=1}^{N} z_k \int_{\partial\Omega} \partial_{x_i}(\gamma_{1,y}^{\mathrm{int}} S(x,y))\psi_k(y)\,\mathrm{d}s_y, \tag{16}$$

for $i = 1,2,3$ together with Hooke's law. If, for instance, the deformations are to be analyzed at several points $x_1, \dots, x_l$, $l \in \mathbb{N}$, this can be understood as the computation of a vector

$$v := \tilde{V}_h y - \tilde{W}_h z \tag{17}$$

with $v = [u_h(x_i)]_{i=1,\dots,l}$. Since the two discrete operators

$$\tilde{V}_h := \left[\int_{\partial\Omega} S(x_i,y)\varphi_j(y)\,\mathrm{d}s_y \right]_{ij} \quad \text{and} \quad \tilde{W}_h := \left[\int_{\partial\Omega} \gamma_{1,y}^{\mathrm{int}} S(x_i,y)\psi_k(y)\,\mathrm{d}s_y \right]_{ik},$$

with $i = 1,\dots,l$, $j = 1,\dots,M$, and $k = 1,\dots,N$ are of collocation type, we are able to accelerate the evaluation of the deformations and stresses with the introduced AMVM. The evaluation of the stresses using the derivatives $\partial_{x_i} u_h$ $i = 1,2,3$, can be done in a similar way using (16).

6. Numerical results

The numerical experiments are divided into two parts. In both cases the numerical solution of the Lamé equations

$$-\mu\,\Delta u(x) - (\lambda + \mu)\,\mathrm{grad}\,\mathrm{div}\,u(x) = 0, \quad x \in \Omega, \tag{18}$$

with the Lamé constants and $E = 1.0\,(N/mm^2)$, $\nu = 0.3$ is computed. The first part deals with the quality of the error estimator in AMVM and the numerical performance of AMVM compared to the multiplication by an approximation obtained from ACA. Then, the numerical performance of the combination of AMVM and BACA adapted to linear elasticity (see Sect. 5) is investigated in comparison with ACA. First calculations of linear elasticity using the ACA were carried out in [6].

The computations in this article were performed on a computer with an Intel(R) Core(TM) i7-6700HQ CPU at 2.60 GHz. All approximation steps in the procedures are performed without parallelization. Furthermore, recompression and agglomeration techniques are applied in the numerical investigations neither to ACA nor to the new methods. Recompression based on QR decomposition as described in [5] can be applied directly for both methods resulting in lower storage requirements. However, a more detailed investigation of how agglomeration can be applied efficiently together with the new methods is required.

In the following tests, the look-ahead approximation $\hat{A}_k$ is two steps of ACA ahead of the current approximation A_k, which turned out to be enough. Notice that this number has to be adapted to the problem in general.

6.1. Quality of AMVM for linear elasticity

The qualitative investigations of AMVM are carried out on four different discretizations of the cube $\Omega = [-1,1]^3$ consisting of 488, 1 946, 7 778, and 31 106 points. The following boundary conditions are chosen

$$\gamma_0^{\mathrm{int}} u(x) = g_D(x) := S(x-p)$$

for $x \in \Gamma_D = \{x \in \Omega : x_1 = 1 \text{ or } x_2 = -1 \text{ or } x_3 = 1\}$ and

$$\gamma_1^{\mathrm{int}} u(x) = g_N(x) := \gamma_1^{\mathrm{int}} S(x-p)$$

for $x \in \Gamma_N = \{x \in \Omega : x_1 = -1 \text{ or } x_2 = 1 \text{ or } x_3 = -1\}$ with $p = (5.0, 5.0, 5.0)^T$. We compare the computational time and the storage requirements of AMVM and ACA when computing the right-hand side of (9). The approximation of the latter will be denoted by b_{AMVM} and b_{ACA}, respectively. The blockwise accuracy of ACA is chosen to be $\varepsilon_{\mathrm{ACA}} = 10^{-6}$ and the admissibility parameter is $\beta = 0.8$. The parameter $b_{\min}$, which is a parameter of the used cluster tree, denotes the minimal size of a matrix block. The results of ACA are presented in Tables 1 and 2.

Applying the adaptive matrix-vector multiplication (AMVM) to the linear elasticity problem described in the beginning of Sect. 6 provides for $\theta = 0.7$ the results shown in Tables 3 and 4. The error $\|b - b_{\mathrm{AMVM}}\|_2$ was kept at the same order of magnitude as $\|b - b_{\mathrm{ACA}}\|_2$ in the previous tests. On all four discretizations of the cube Ω a reduction of the computational time could be achieved. The

Table 2
Storage requirements for the approximations constructed by ACA.

N	M	$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}	
		MB	%	MB	%	MB	%	MB	%	MB	%
972	488	3.0	83.5	3.5	95.6	3.5	96.5	3.5	96.4	3.4	95.5
3 888	1 946	19.7	34.1	23.9	41.4	23.4	40.5	23.3	40.4	23.8	41.3
15 552	7 778	115.6	12.5	137.6	14.9	132.7	14.4	131.3	14.2	137.7	14.9
62 208	31 106	870.0	5.9	993.6	6.7	930.1	6.3	900.5	6.1	974.3	6.6

N	M	V_{23}		V_{33}		$K_{\Delta,h}$	
		MB	%	MB	%	MB	%
972	488	3.5	96.1	3.5	95.6	3.6	99.6
3 888	1 946	23.3	40.4	23.8	41.3	31.1	53.9
15 552	7 778	131.4	14.2	136.2	14.8	201.1	21.8
62 208	31 106	911.7	6.1	970.1	6.6	1328.9	9.3

Table 3
Error and time required to compute right-hand side of (9) via AMVM.

N	M	$b_{\min}$	$\|b - b_{\text{AMVM}}\|_2$	time approximation
972	488	15	$4.69 \cdot 10^{-7}$	5.5 s
3 888	1 946	20	$3.49 \cdot 10^{-7}$	29.9 s
15 552	7 778	30	$2.81 \cdot 10^{-7}$	179.6 s
62 208	31 106	40	$3.35 \cdot 10^{-7}$	1 097.1 s

Table 4
Storage requirements for the approximations constructed by AMVM.

N	M	$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}	
		MB	%	MB	%	MB	%	MB	%	MB	%
972	488	2.9	79.1	3.0	84.2	3.0	84.2	3.0	84.4	3.0	84.2
3888	1 946	19.3	33.5	21.3	37.0	20.9	36.2	20.8	36.1	21.3	36.9
15 552	7 778	114.5	12.4	124.7	13.5	120.5	13.0	119.5	12.9	124.9	13.5
62 208	31 106	838.6	5.7	919.5	6.2	836.8	5.7	811.9	5.5	915.2	6.2

N	M	V_{23}		V_{33}		$K_{\Delta,h}$	
		MB	%	MB	%	MB	%
972	488	3.0	84.3	3.0	84.2	2.4	68.0
3888	1 946	20.8	36.1	21.3	36.9	17.2	29.9
15 552	7 778	119.6	12.9	119.4	14.8	116.1	12.6
62 208	31 106	822.1	5.5	902.5	6.1	803.3	5.6

Table 5
Required time to compute right-hand side of (9) via ACA and AMVM using $b_{\min} = 40$.

N	M	time approximation ACA	time approximation AMVM
972	488	5.4 s	5.1 s
3 888	1 946	36.9 s	29.4 s
15 552	7 778	232.9 s	181.9 s
62 208	31 106	1428.6 s	1 097.1 s

storage requirements of the operators $V_{\Delta,h}$, V_{11}, V_{12}, V_{13}, V_{22}, V_{23}, and V_{33} turn out to be slightly lower than the corresponding approximations obtained via ACA. The main benefit is obtained for the operator $K_{\Delta,h}$.

Tables 1 and 3 give the impression that the logarithmic-linear complexity is not fulfilled. This can be explained by the different parameters $b_{\min}$. Table 5 shows the approximation times for a constant $b_{\min}$ and a comparable error $\|b - b_{\text{AMVM}}\|_2$, where we can observe that the runtime approaches logarithmic-linear complexity.

Before moving on to a more realistic problem, we take a closer look at the reliability and efficiency of the error estimator. We present the results obtained in the case of the discretization consisting of 488 points and 972 mesh elements and 7 778 points and 15 552 mesh elements. We employ a rank-2 approximation to start the iterative approximation process. Fig. 2 shows that the error

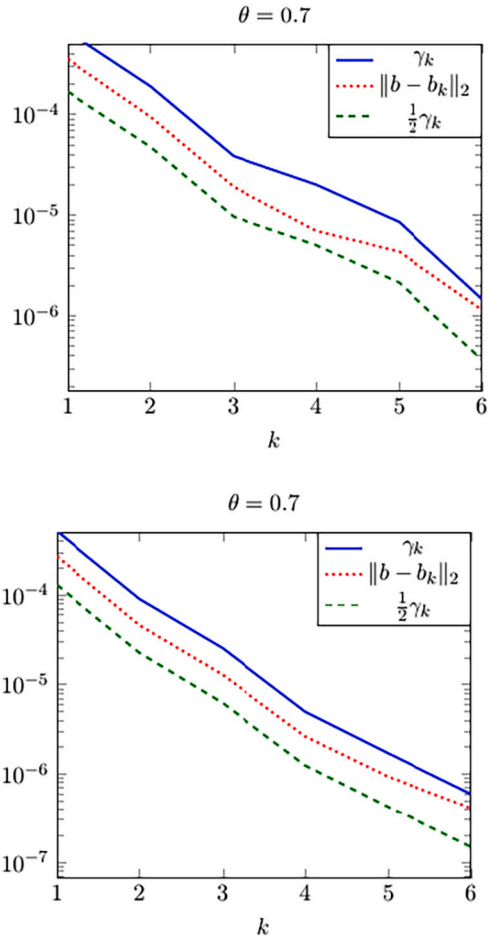

Fig. 2. Quality of the error estimator γ_k in the case of AMVM (top: 488 points, bottom 7778 points). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

estimator γ_k estimates the error $\|b - b_k\|_2$ of the right-hand side reliably and efficiently, which confirms the theoretical results of the adaptive matrix-vector multiplication presented in Sect. 3.

6.2. Beam with double-T shape: load in z-direction

The following experiments focus on the numerical solution of the Lamé equations on three discretizations of the geometry shown in Fig. 3. The beam has a length, height and width of 2 with a central part having height and width of 1. Fig. 4 shows the assignment of the boundary elements to Dirichlet and Neumann part. On the blue area the beam is loaded with a force of 0.1 N, while the Dirichlet boundary is illustrated by the green area. On the remaining part of the boundary, i.e. on the gray area in Fig. 4, homogeneous Neumann boundary conditions ($\gamma_1^{\text{int}} u(x) = 0$) are prescribed. The right-hand side of the system of equations which has to be computed is obtained by multiplying the given boundary data by the respective discretized operators V_h, K_h and D_h; cf. (9).

We compare the approximate solution obtained from approximating the coefficient matrix via BACA and ACA, respectively.

The deformations of the beam under load in z-direction are shown in Fig. 5. The maximum absolute differences between the deformations generated via ACA and BACA in x-, y- and z-direction are $1.2 \cdot 10^{-4}$, $2.4 \cdot 10^{-4}$ and $3.3 \cdot 10^{-4}$. So, both methods ACA and BACA give similar results.

The parameters used for ACA in Sect. 6.1 remain unchanged. Additionally, we use $\varepsilon_{\text{BPCG}} = 10^{-5}$ which denotes the accuracy of the Bramble-Pasciak conjugate gradient method [10] during the iterative solution procedure. The results for ACA are shown in Table 6.

For BACA other parameters have to be chosen. The adaptive adjustment of the error tolerance in the Bramble-Pasciak CG is done according to condition (14) with $\alpha = 10$, see 2. of Algorithm 3. The initial value of the accuracy in Bramble-Pasciak CG is 10^{-1}. The tolerance $\varepsilon_{\text{BACA}}$ is 10^{-4} and $\theta = 0.8$. The starting approximations of the respective V operators are obtained by applying 8 (for the two coarsest grids) and 10 (for the finest grid) ACA steps. For the operator K the respective number of steps are 4 and 6. Solving the Lamé equations via BACA with these parameters leads after four iteration steps to the values shown in Table 7.

Compared to the results obtained from ACA, no significant differences can be observed when applying BACA to the operator $V_{\Delta,h}$. The V operators require only about 70–80% of the storage needed for the approximations generated via ACA. Stronger benefits can

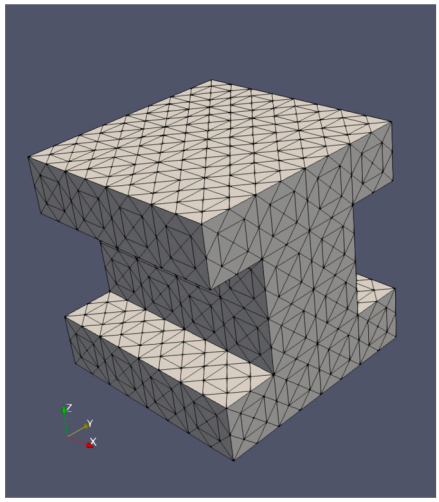

Fig. 3. Discretization of double T-beam.

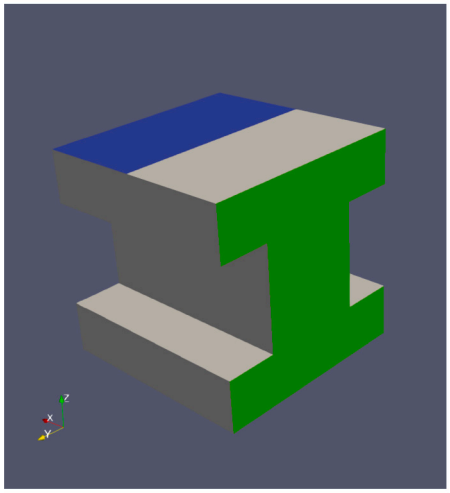

Fig. 4. Dirichlet boundary green and loaded Neumann boundary part blue.

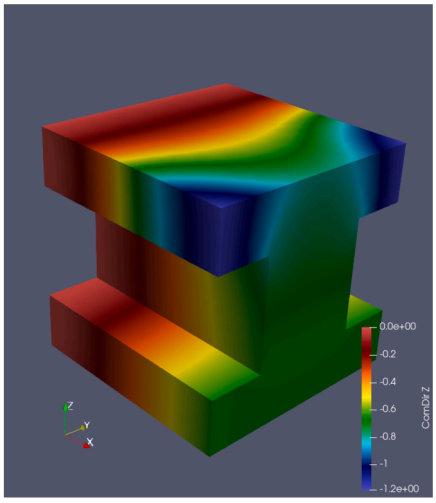

Fig. 5. Deformation under loading in z-direction for ACA using a mesh with 3330 nodes and 6656 elements.

Table 6
Storage requirements of the approximations constructed via ACA and time consumption of solving the problem.

N	M	$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}
		MB	%	MB	%	MB	%	MB	%	MB
1 664	834	6.9	65.2	8.3	78.3	8.4	79.8	8.3	78.7	8.4
6 656	3 330	44.4	26.2	56.3	33.3	56.5	33.4	54.6	32.3	56.3
26 624	13 314	238.7	8.8	306.5	11.3	300.8	11.1	285.3	10.6	305.1

N	M	V_{22}	V_{23}		V_{33}		$K_{\Delta,h}$		approximation
		%	MB	%	MB	%	MB	%	time
1 664	834	79.7	8.4	79.4	8.2	77.2	9.5	90.1	15.1 s
6 656	3 330	33.3	54.6	32.3	54.0	31.9	75.2	44.9	91.2 s
26 624	13 314	11.3	283.7	10.5	287.2	10.6	497.3	18.4	546.4 s

Table 7
Storage, relative storage for the approximations constructed by BACA and time consumption of solving the problem after applying BACA in the case of Lamé equations.

N	M	$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}
		MB	%	MB	%	MB	%	MB	%	MB
1 664	834	6.5	61.0	6.8	64.2	6.8	64.4	6.8	63.9	6.8
6 656	3 330	40.1	23.7	41.5	24.5	41.2	24.4	40.2	23.8	41.5
26 624	13 314	220.3	8.1	228.9	8.5	223.8	8.3	213.2	7.9	228.8

N	M	V_{22}	V_{23}		V_{33}		$K_{\Delta,h}$		approximation
		%	MB	%	MB	%	MB	%	time
1 664	834	64.5	6.8	64.0	6.8	63.9	4.7	44.4	10.1 s
6 656	3 330	24.6	40.2	23.8	40.5	24.0	27.3	16.1	48.1 s
26 624	13 314	8.5	212.7	7.9	218.3	8.1	180.4	6.7	287.6 s

be achieved for the K_Δ operator. Here, the approximation using BACA requires only 50% (for the coarsest grid) and 36% (for the two finest grids) of the storage needed in the case of ACA. Table 7 also shows advantages of BACA with respect to the approximation time. While on the coarsest grid 67% of the time needed by ACA is consumed, for the second and third finest grid the time can be reduced to 53%. Note that this is a reduction in computation time and storage requirements compared to those methods that already have linear-logarithmic complexity.

The previous numerical example with the used parameters is still a rather academic example. Using the parameters of steel, i.e. $E = 210\,000\,N/mm^2$, $\nu = 0.28$, and a load of $10\,kN$ leads to a maximum deformation of the beam in z-direction of $0.072\,mm$ for both methods on the grid with $N = 6\,656$ and $M = 3\,330$. The computation time of 261 seconds for BACA is significantly reduced compared to ACA, which required 453 seconds.

Since the saturation assumption (13) is required for the reliability of the error estimator $\mathcal{E}_k$, this assumption is checked for the above numerical example. Denoting

$$c_k = \frac{\|\hat{A}_k x_k - A x_k\|_2}{\|A_k x_k - A x_k\|_2},$$

for the performed iteration steps we get $c_1 = 0.48$, $c_2 = 0.61$, and $c_3 = 0.64$ for the smallest grid in the case of rank-1 updates. For rank-2 updates these values become $c_1 = 0.19$, $c_2 = 0.36$ and $c_1 = 0.1$, $c_2 = 0.23$ in the case of rank-3 updates.

7. Conclusion

In this article, a new method for an adaptive and approximate computation of a matrix-vector multiplication was presented for the case of discretizations of integral operators. The goal was to adapt the approximation to the structure of the vector to be multiplied in order to reduce the storage requirements of the matrix as well as the computational time. Techniques known from adaptive mesh refinement were used in order to identify those blocks which are important for the error of the multiplication.

After analyzing the convergence of the adaptive method, we focused on the Lamé equations as an application example. Therefore, the adaptation of the new method in the case of linear elasticity was discussed and performed for both approximating the system matrix on the left-hand side and approximating the action of operators on the right-hand side. In the numerical examples, the quality of the employed estimator, i.e. its reliability and efficiency, could be observed.

The application of the new methods in case of a loaded beam with double-T shape resulted in less storage requirements and a significant reduction of the computation time compared to solving the considered problem using ACA.

References

[1] A. Aurada, S. Ferraz-Leite, D. Praetorius, Estimator reduction and convergence of adaptive BEM, Appl. Numer. Math. 62 (2012) 787–801.
[2] M. Bauer, M. Bebendorf, Block-adaptive cross approximation of discrete integral operators, Comput. Methods Appl. Math. 21 (1) (2021) 13–29.
[3] M. Bauer, M. Bebendorf, B. Feist, Kernel-independent adaptive construction of $\mathcal{H}^2$-matrix approximations, Numer. Math. (2021), https://doi.org/10.1007/s00211-021-01255-y.
[4] M. Bebendorf, Approximation of boundary element matrices, Numer. Math. 86 (2000) 565–589.
[5] M. Bebendorf, Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems, Lecture Notes in Computational Science and Engineering (LNCSE), vol. 63, Springer, Berlin, 2008.
[6] M. Bebendorf, R. Grzhibovskis, Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation, Math. Methods Appl. Sci. 29 (2006) 1721–1747.
[7] M. Bebendorf, S. Rjasanow, Adaptive low-rank approximation of collocation matrices, Computing 70 (2003) 1–24.
[8] S. Börm, W. Hackbusch, $\mathcal{H}^2$-matrix approximation of integral operators by interpolation, Appl. Numer. Math. 43 (2002) 139–143.
[9] S. Börm, M. Löhndorf, J.M. Melenk, Approximation of integral operators by variable-order interpolation, Numer. Math. 99 (4) (2005) 605–643.
[10] J.H. Bramble, J.E. Pasciak, A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems, Math. Comput. 50 (181) (1988) 1–17.
[11] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, SIAM J. Numer. Anal. 33 (1996) 1106–1124.
[12] G. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrica 1 (1936) 211–218.
[13] S. Ferraz-Leite, D. Praetorius, Simple a posteriori error estimators for the h-version of the boundary element method, Computing 83 (2008) 135–162.
[14] S.A. Goreinov, E.E. Tyrtyshnikov, N.L. Zamarashkin, A theory of pseudoskeleton approximations, Linear Algebra Appl. 261 (1997) 1–21.
[15] L. Grasedyck, W. Hackbusch, Constructions and arithmetics of $\mathcal{H}$-matrices, Computing 70 (2003) 295–334.
[16] L.F. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.
[17] W. Hackbusch, A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices, Computing 62 (1999) 89–108.
[18] W. Hackbusch, B.N. Khoromskij, A sparse $\mathcal{H}$-matrix arithmetic. Part II: application to multi - dimensional problems, Computing 64 (2000) 21–47.
[19] H. Han, The boundary integro-differential equations of three dimensional Neumann problem in linear elasticity, Numer. Math. 68 (2) (1994) 269–281.
[20] R.B. Hetnarski, J. Ignaczak, Mathematical Theory of Elasticity, Taylor & Francis, 2004.
[21] S. Rjasanow, O. Steinbach, The Fast Solution of Boundary Integral Equations, Mathematical and Analytical Techniques with Applications to Engineering, Springer, New York, 2007.
[22] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. 60 (2) (1985) 187–207.
[23] S.A. Sauter, C. Schwab, Boundary Element Methods, Springer Series in Computational Mathematics, Springer, Berlin, 2011.
[24] O. Steinbach, Numerical Approximation Methods for Elliptic Boundary Value Problems, Springer, New York, 2008.
[25] P.P. Teodrescu, Treatise on Classical Elasticity. Theory and Related Problems, Springer, 2013.