# Max-Planck-Institut
## für Mathematik
## in den Naturwissenschaften
## Leipzig


**Stabilised rounded addition of hierarchical matrices**

by


*Mario Bebendorf, and Wolfgang Hackbusch*



Preprint no.: 4 2007


# Stabilised rounded addition of hierarchical matrices

M. Bebendorf*and W. Hackbusch†

The efficiency of hierarchical matrices is based on the approximate evaluation of usual matrix operations. The introduced approximation error may, however, lead to a loss of important matrix properties. In this article we present a technique which preserves the positive definitness of a matrix independently of the approximation quality. The importance of this technique is illustrated by an elliptic mixed boundary value problem with tiny Dirichlet part.

## 1 Introduction

The technique of hierarchical matrices allows to perform the standard matrix operations ($Ax$, $A+B$, $A*B$, $A^{-1}$, $LU$ decomposition), provided that the matrices are originating from elliptic boundary value problems or corresponding integral operators. Since the storage and CPU time are proportional to $n\log^q n$, where $n$ denotes the matrix size, these method can compete with standard iterative solvers in the case of solving linear systems. However, these operations are not exact but approximate. The involved error depends on the choice of the local rank $k$. Since the accuracy improves exponentially with increasing $k$, a moderate choice $k\sim\log(1/\varepsilon)$ suffices to reach a given tolerance $\varepsilon$.

Depending of the choice of $k$, there are two different strategies of using the hierarchical matrix technique:

(i) We may achieve a high accuracy by a larger choice of $k$. Then the technique of hierarchical matrices can be considered as a direct solution method. Of course, the computational work increases with $k$.

(ii) It is cheaper to perform the calculations with rather small $k$. Then the accuracy is low, but may even be sufficient to produce a good preconditioner. In that case, there is a further coarsening of the hierarchical block structure which reduces the computational work; see Sect. 5.

Although the method of hierarchical matrices works astonishingly robust, there are some ill-conditioned problems, where small perturbations may turn, e.g., a positive definite matrix into a indefinite one so that a following Cholesky decomposition fails. In principle, a larger choice of $k$ can avoid the problem. However, as stated in version (ii), we deliberately like to use a lower accuracy in order to reduce the computational costs.

The principal source of approximation errors is the truncation of a matrix block to lower rank. Let the exact matrix block of rank $\ell$ be represented by its singular value decomposition $\sum_{i=1}^{\ell}\sigma_i u_i v_i^H$. A

*Fakultät für Mathematik und Informatik, Universität Leipzig, Johannisgasse 26, 04103 Leipzig
†Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstraße 22, 04103 Leipzig

truncation to rank $k < \ell$ yields $\sum_{i=1}^{k} \sigma_i u_i v_i^H$ as a best approximation. The truncation consists of $\ell - k$ subtractions of rank-1 matrices $\sigma_i u_i v_i^H$. In this paper we propose to compensate each subtraction of a rank-1 matrix by a modification on the diagonal. Then it can be guaranteed that a positive definite input matrix remains positive definite during the truncation process. A numerical example is given in Sect. 6 which is extremely sensible to perturbations, since the lowest eigenvalue is quite close to zero. We show that the standard approach fails for the desired choice of a low rank $k$, while the method proposed here works robustly.

The coarsening process mentioned above requires several local singular value decompositions. The cost is moderate since $k$ is not large, but the number of operations can be reduced by a factor of almost 2, if we replace the standard representation $AB^H$ of low-rank matrices by their singular value representations. The particular advantages in connection with the coarsening process are explained in Sect. 4.

## 2 Hierarchical Matrices

The structure of hierarchical matrices was first introduced in [14, 16]. Let $I$ and $J$ be index sets and assume that a partition $P$ of $I \times J$ has been generated such that each block satisfies a so-called admissibility condition, which is the characteristic property for the existence of low-rank approximants to the respective sub-block of $A \in \mathbb{C}^{I \times J}$. The construction of $P$ is usually done by recursive subdivision of the row and column indices $I$ and $J$. A subdivision strategy based on the *principal component analysis* is presented in [1], a method which gives similar results was proposed in [9]. The resulting *cluster trees* will be denoted by $T_I$ and $T_J$. The construction of $T_I$ at least in the case of quasi-uniform grids has complexity $\mathcal{O}(|I| \log |I|)$. The depth of $T_I$, i.e., the maximum level of a vertex in $T_I$ increased by one, will be denoted by $d(T_I)$, while $S(t)$ will be used as the set of sons of a vertex $t \in T_I$. For reasonable cluster trees $T_I$ one expects $d(T_I) \sim \log_2 |I|$. The recursive subdivision stops at a cluster $t$ whenever its cardinality $|t|$ falls below a given number $n_{\min} \in \mathbb{N}$. $\mathcal{L}(T_I)$ stands for the set of leaves of $T_I$.

Using the cluster trees $T_I$ and $T_J$, which embody a hierarchy of partitions of $I$ and $J$, $P := \mathcal{L}(T_{I\times J})$ is defined as the leaves of the so-called *block cluster tree* $T_{I\times J}$, which contains a hierarchy of partitions of $I \times J$. The block cluster tree terminates at blocks $t \times s$, $t \in T_I$, $s \in T_J$, which satisfy a condition that guarantees the existence of low-rank approximants. All other blocks are refined by subdividing the clusters $t$ and $s$ according to their cluster trees. The complexity of building $T_{I\times J}$ in the case of quasi-uniform grids can be estimated to be of the order $\mathcal{O}(|I| \log |I| + |J| \log |J|)$; cf. [1, 13]. An important property of the used hierarchical partitions is that for a given $t \in T_I$ or a given $s \in T_J$ a constantly bounded number of blocks $t \times s$ appear in $T_{I\times J}$; i.e., the expressions

$$c_{\mathrm{sp}}^r := \max_{t\in T_I} |\{s \subset J : t \times s \in T_{I\times J}\}| \quad \text{and} \quad c_{\mathrm{sp}}^c := \max_{s\in T_J} |\{t \subset I : t \times s \in T_{I\times J}\}|$$

are bounded independently of the sizes of $I$ and $J$; see [13]. As a consequence, we obtain an estimate for the following expression which will appear in many complexity estimates such as the storage requirement of an $\mathcal{H}$-matrix

$$\sum_{t\times s\in T_{I\times J}} [|t| + |s|] = \sum_{t\in T_I} \sum_{t\times s\in T_{I\times J}} |t| + \sum_{s\in T_J} \sum_{t\times s\in T_{I\times J}} |s| \tag{1a}$$

$$\leq c_{\mathrm{sp}} \left( \sum_{t\in T_I} |t| + \sum_{s\in T_J} |s| \right) \leq c_{\mathrm{sp}} d(T_{I\times J})[|I| + |J|], \tag{1b}$$

where $c_{\mathrm{sp}} := \max\{c_{\mathrm{sp}}^r, c_{\mathrm{sp}}^c\}$. A further consequence of the hierarchical structure is the following lemma; see [11].

**Lemma 2.1.** *Let $P$ be a partition as described above. Then for any matrix $M \in \mathbb{R}^{n\times n}$ the following inequality holds between the global and the blockwise spectral norms:*

$$\|A\|_2 \leq c_{\mathrm{sp}} d(T_{I\times J}) \max_{b\in P} \|A_b\|_2, \quad \text{where } A_b = (a_{ij})_{i\in t,\, j\in s} \text{ for } b = t \times s.$$

Using the previous notation, we can define the set of $\mathcal{H}$-matrices for a block cluster tree $T_{I\times J}$ with blockwise rank $k$:

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

Note that $\mathcal{H}(T_{I\times J}, k)$ is not a linear space, since the sum of two matrices from

$$\mathbb{C}_k^{m\times n} := \{A \in \mathbb{C}^{m\times n} : \operatorname{rank} A \leq k\}$$

exceeds rank $k$ in general. If, however, two matrices $A \in \mathbb{C}_{k_A}^{m\times n}$ and $B \in \mathbb{C}_{k_B}^{m\times n}$ having the representations $A = U_A V_A^H$, $U_A \in \mathbb{C}^{m\times k_A}$, $V_A \in \mathbb{C}^{n\times k_A}$, and $B = U_B V_B^H$, $U_B \in \mathbb{C}^{m\times k_B}$, $V_B \in \mathbb{C}^{n\times k_B}$, are to be added, the sum

$$A + B = [U_A, U_B][V_A, V_B]^H$$

might be close to a matrix of a much smaller rank. In this case, the *rounded addition* of two low-rank matrices, which truncates the sum to rank $k$, can be used. It is known (see [8, 18]) that the closest matrix in $\mathbb{C}_k^{m\times n}$ to a given matrix from $A \in \mathbb{C}^{m\times n}$, $m \geq n$, is given by the sum of the major $k$ singular triplets; i.e, let $A = U\Sigma V^H$ be a singular value decomposition of $A$, then for $k \in \mathbb{N}$, $k \leq n$, it holds that

$$\min_{M\in\mathbb{C}_k^{m\times n}} \|A - M\| = \|A - A_k\| = \|\Sigma - \Sigma_k\|, \tag{2}$$

where $A_k := U\Sigma_k V^H \in \mathbb{C}_k^{m\times n}$ and $\Sigma_k := \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0) \in \mathbb{R}^{n\times n}$. Here, $\|\cdot\|$ is any unitarily invariant norm.

The information about the error $\|A - A_k\| = \|\Sigma - \Sigma_k\|$ can be used in two different ways. If the maximum rank $k$ of the approximant is prescribed, one gets to know the associated error. If on the other hand a relative accuracy $\varepsilon > 0$ of the approximant $A_k$ is prescribed (say with respect to the spectral norm), i.e.,

$$\|A - A_k\|_2 < \varepsilon \|A\|_2,$$

then due to (2) the required rank $k(\varepsilon)$ is given by

$$k(\varepsilon) := \min\{k \in \mathbb{N} : \sigma_{k+1} < \varepsilon\sigma_1\}.$$

Using the rounded addition on each sub-block of an $\mathcal{H}$-matrix, it is therefore possible to define an approximate matrix addition under which $\mathcal{H}(T_{I\times J}, k)$ is closed. Based on this substitute of the usual addition, approximate matrix operations with logarithmic-linear complexity can be defined; see [14, 16, 15, 13]. Hierarchical $LU$ decompositions (see [3]) can be used for preconditioning linear systems. In addition to the efficient treatment of discretisations of integral operators (see [7]), theoretical evidence for the applicability of $\mathcal{H}$-matrices was given when approximating inverses of finite-element stiffness matrices for general elliptic second order partial differential operators; see [6, 2]. The approximation of the factors of the $LU$ decomposition of finite-element stiffness matrices was investigated in [4].

Although the introduced approximation error allows to guarantee almost linear complexity, it comes with the disadvantage that important matrix properties which are present in exact arithmetic might be lost.

## 3 Preserving positivity

The acceleration of matrix operations by the $\mathcal{H}$-matrix arithmetic is connected with a certain approximation error, which in particular perturbs the eigenvalues of the results of these operations. If the smallest eigenvalue is close to the origin compared with the rounding accuracy, it may happen that the result of these operations becomes indefinite although it should be positive definite in exact arithmetic. Since the introduced error arises only from the rounded addition of low-rank matrices, we concentrate on the rounded addition in order to avoid the loss of positivity.

Assume that $\hat{A} \in \mathbb{C}^{I\times I}$ is the Hermitian positive definite result of an exact addition of two matrices from $\mathcal{H}(T_{I\times I}, k)$ and let $A \in \mathcal{H}(T_{I\times I}, k)$ be its $\mathcal{H}$-matrix approximant. For a moment we assume that $\hat{A}$ and $A$ differ only on a single off-diagonal block $t \times s \in P$. Let $EF^H$, $E \in \mathbb{C}^{t\times k}$, $F \in \mathbb{C}^{s\times k}$, be the error matrix associated with $t \times s$; i.e.,

$$A_{ts} = \hat{A}_{ts} - EF^H$$

and let

$$\varepsilon := \max\{\|E\|_2^2, \|F\|_2^2\}. \tag{3}$$

Due to symmetry, $FE^H$ is the error matrix on block $s \times t$.

We modify the approximant $A$ in such a way that the new approximant $\tilde{A}$ is guaranteed to be positive definite. This is done by adding $EE^H$ to $A_{tt}$ and $FF^H$ to $A_{ss}$ such that

$$\begin{bmatrix} \tilde{A}_{tt} & \tilde{A}_{ts} \\ \tilde{A}_{ts}^H & \tilde{A}_{ss} \end{bmatrix} := \begin{bmatrix} A_{tt} & A_{ts} \\ A_{ts}^H & A_{ss} \end{bmatrix} + \begin{bmatrix} EE^H & \\ & FF^H \end{bmatrix} = \begin{bmatrix} \hat{A}_{tt} & \hat{A}_{ts} \\ \hat{A}_{ts}^H & \hat{A}_{ss} \end{bmatrix} + \begin{bmatrix} EE^H & -EF^H \\ -FE^H & FF^H \end{bmatrix}.$$

Since

$$\begin{bmatrix} EE^H & -EF^H \\ -FE^H & FF^H \end{bmatrix} = \begin{bmatrix} -E \\ F \end{bmatrix} \begin{bmatrix} -E \\ F \end{bmatrix}^H$$

is positive semi-definite, the eigenvalues of $\tilde{A}$ are not smaller than those of $\hat{A}$. Therefore, $\tilde{A}$ is Hermitian positive definite and

$$\| \begin{bmatrix} \tilde{A}_{tt} & \tilde{A}_{ts} \\ \tilde{A}_{ts}^H & \tilde{A}_{ss} \end{bmatrix} - \begin{bmatrix} \hat{A}_{tt} & \hat{A}_{ts} \\ \hat{A}_{ts}^H & \hat{A}_{ss} \end{bmatrix} \|_2 = \| \begin{bmatrix} -E \\ F \end{bmatrix} \|_2^2 \leq \|E\|_2^2 + \|F\|_2^2 \leq 2\varepsilon.$$

A relative error estimate can be obtained if we assume that

$$\|E\|_2^2 \leq \varepsilon \|\hat{A}_{tt}\|_2, \quad \|F\|_2^2 \leq \varepsilon \|\hat{A}_{ss}\|_2, \quad \text{and} \quad \|EF^H\|_2 \leq \varepsilon \|\hat{A}_{ts}\|_2$$

holds instead of (3). In this case, we obtain

$$\| \begin{bmatrix} \tilde{A}_{tt} & \tilde{A}_{ts} \\ \tilde{A}_{ts}^H & \tilde{A}_{ss} \end{bmatrix} - \begin{bmatrix} \hat{A}_{tt} & \hat{A}_{ts} \\ \hat{A}_{ts}^H & \hat{A}_{ss} \end{bmatrix} \|_2 = \| \begin{bmatrix} EE^H & -EF^H \\ -FE^H & FF^H \end{bmatrix} \|_2 \leq 2\varepsilon \| \begin{bmatrix} \hat{A}_{tt} & \hat{A}_{ts} \\ \hat{A}_{ts}^H & \hat{A}_{ss} \end{bmatrix} \|_2,$$

which follows from the fact that

$$\left\| \begin{bmatrix} A \\ B \end{bmatrix} \right\|_2 \leq \sqrt{2} \left\| \begin{bmatrix} C \\ D \end{bmatrix} \right\|_2 \quad \text{and} \quad \| \begin{bmatrix} A & B \end{bmatrix} \|_2 = \left\| \begin{bmatrix} A^T \\ B^T \end{bmatrix} \right\|_2 \leq \sqrt{2} \left\| \begin{bmatrix} C^T \\ D^T \end{bmatrix} \right\|_2 = \sqrt{2} \| \begin{bmatrix} C & D \end{bmatrix} \|_2$$

for matrices $A$, $B$, $C$, and $D$ satisfying $\|A\|_2 \leq \|C\|_2$ and $\|B\|_2 \leq \|D\|_2$. The last estimates are a consequence of

$$\begin{aligned} \left\| \begin{bmatrix} A \\ B \end{bmatrix} \right\|_2^2 &\leq 2 \sup_{x\neq 0} \max \left\{ \frac{\|Ax\|_2^2}{\|x\|_2^2}, \frac{\|Bx\|_2^2}{\|x\|_2^2} \right\} = 2 \max\{\|A\|_2^2, \|B\|_2^2\} \leq 2 \max\{\|C\|_2^2, \|D\|_2^2\} \\ &= 2 \sup_{x\neq 0} \max \left\{ \frac{\|Cx\|_2^2}{\|x\|_2^2}, \frac{\|Dx\|_2^2}{\|x\|_2^2} \right\} \leq 2 \sup_{x\neq 0} \frac{\|Cx\|_2^2 + \|Dx\|_2^2}{\|x\|_2^2} = 2 \left\| \begin{bmatrix} C \\ D \end{bmatrix} \right\|_2^2. \end{aligned}$$

Since $t \times t$ and $s \times s$ will usually not be leaves in $T_{I\times I}$, it is necessary that $EE^H$ and $FF^H$ are restricted to the leaves contained in $t \times t$ and $s \times s$ when adding them to $A_{tt}$ and $A_{ss}$, respectively. Note that this leads to rounding errors which in turn have to be added to the diagonal sub-blocks of $t \times t$ and $s \times s$ in order to preserve positivity. The computational complexity which is connected with the rounded addition makes it necessary to remodify the above idea. Once again, we replace an approximant with another approximant by adding a positive semi-definite matrix. Let $t_1$ and $t_2$ be the sons of $t$ and let $s_1$ and $s_2$ be the sons of $s$. If we define

$$\tilde{\tilde{A}}_{tt} := \tilde{A}_{tt} + \begin{bmatrix} E_{t_1} \\ E_{t_2} \end{bmatrix} \begin{bmatrix} E_{t_1} \\ E_{t_2} \end{bmatrix}^H = A_{tt} + \begin{bmatrix} 2E_{t_1}E_{t_1}^H & 0 \\ 0 & 2E_{t_2}E_{t_2}^H \end{bmatrix},$$

the problem of adding $EE^H$ to $A_{tt}$ is reduced to adding $2E_{t_1}E_{t_1}^H$ to $A_{t_1t_1}$ and $2E_{t_2}E_{t_2}^H$ to $A_{t_2t_2}$. Applying this idea recursively, adding $EE^H$ to $A_{tt}$ can finally be done by adding a multiple of $E_{t^*}E_{t^*}^H$ to the dense matrix block $A_{t^*t^*}$ for each leaf $t^*$ in $T_I$ from the set of descendants of $t$. We obtain the following two algorithms addsym_stab and addsym_diag; see Algorithm 3.1 and Algorithm 3.2.

```
procedure addsym_stab(t, s, U, V, var A)
if t × s is non-admissible then
    add UV^H to A_ts without approximation;
else
    add UV^H to A_ts using the rounded addition;
    denote by EF^H the rounding error;
    addsym_diag(t, E, A);
    addsym_diag(s, F, A);
endif
```

Algorithm 3.1: Stabilised Hermitian rounded addition

The first algorithm adds a matrix of low rank $UV^H$ to an off-diagonal block $t \times s$, $t \neq s$, while the latter adds $EE^H$ to the diagonal block $t \times t$. Note that we assume that an Hermitian matrix is represented by its upper-triangular part only.

```
procedure addsym_diag(t, E, var A)
if t × t is a leaf then
    add EE^H to A_tt without approximation;
else
    addsym_diag(t_1, √2 E_{t_1}, A);
    addsym_diag(t_2, √2 E_{t_2}, A);
endif
```

Algorithm 3.2: Stabilised diagonal addition

We will now estimate the costs if the above algorithms are applied to $t \times s \in T_{I\times I}$. Denote by $N_{\text{diag}}^{\text{stab}}(t)$ the number of operations needed if Algorithm 3.2 is applied to $t \in T_I$ with $E \in \mathbb{C}^{t\times k}$. Since

$$N_{\text{diag}}^{\text{stab}}(t) = N_{\text{diag}}^{\text{stab}}(t_1) + N_{\text{diag}}^{\text{stab}}(t_2)$$

and since $k|t^*|^2/2$ operations are required on each leaf $t^*$ of $T_t$, we obtain

$$N_{\text{diag}}^{\text{stab}}(t) = \sum_{t^*\in\mathcal{L}(T_t)} N_{\text{diag}}^{\text{stab}}(t^*) \leq \sum_{t^*\in\mathcal{L}(T_t)} \frac{k}{2}|t^*|^2 \leq \frac{n_{\min}k}{2} \sum_{t^*\in\mathcal{L}(T_t)} |t^*| = \frac{n_{\min}k}{2}\,|t|.$$

Additionally, denote by $N_{\text{add}}^{\text{stab}}(t,s)$ the number of operations required for adding $UV^H \in \mathbb{C}_k^{t\times s}$ to $A_{ts}$ using Algorithm 3.1. If $t \times s$ is non-admissible, then $\min\{|t|,|s|\} \leq n_{\min}$, which leads to

$$N_{\text{add}}^{\text{stab}}(t,s) \leq |t||s| \leq n_{\min}(|t| + |s|).$$

In the other case, $t \times s$ is admissible. Since for $t \times s \in T_{I\times I}$ a rounded addition and two calls of addsym_diag have to be performed, the costs of Algorithm 3.2 can be estimated by

$$\begin{aligned} N_{\mathrm{add}}^{\mathrm{stab}}(t,s) &= \max\{k^2, n_{\min}\}(|t|+|s|) + N_{\mathrm{diag}}^{\mathrm{stab}}(t) + N_{\mathrm{diag}}^{\mathrm{stab}}(s) \\ &\leq [\max\{k^2, n_{\min}\} + n_{\min}k/2](|t|+|s|). \end{aligned}$$

Hence, the stabilised addition has asymptotically the same computational complexity as the rounded addition on each block.

If two $\mathcal{H}$-matrices are two be added, the stabilised rounded addition has to be applied to each block. The resulting $\mathcal{H}$-matrix will differ from the result $S_{\mathcal{H}}$ of the rounded addition only in the diagonal blocks of $P$. Hence, it requires the same amount of storage. The following theorem gathers the estimates of this section.

**Theorem 3.1.** *Let $A, B$ be Hermitian and let $\lambda_i$, $i \in I$, denote the eigenvalues of $A+B$. Assume that $S_{\mathcal{H}} \in \mathcal{H}(T_{I\times I}, k)$ has precision $\varepsilon$. Using the stabilised rounded addition on each block leads to a matrix $\tilde{S}_{\mathcal{H}} \in \mathcal{H}(T_{I\times I}, k)$ with eigenvalues $\tilde{\lambda}_i \geq \lambda_i$, $i \in I$, satisfying*

$$\|A + B - \tilde{S}_{\mathcal{H}}\|_2 \sim d(T_I)|I|\varepsilon.$$

*Hence, if $A+B$ is positive definite, so is $\tilde{S}_{\mathcal{H}}$. At most $[2\max\{k^2, n_{\min}\} + n_{\min}k]d(T_I)|I|$ operations are required for the construction of $\tilde{S}_{\mathcal{H}}$.*

*Proof.* Let $t^* \in \mathcal{L}(T_I)$ and let $t \in T_I$ be an ancestor of $t^*$ from the $\ell$-th level of $T_I$. Since at most $c_{\mathrm{sp}}$ blocks $t \times s$, $s \in T_I$, are contained in $P$, addsym_diag is applied to $t$ only $c_{\mathrm{sp}}$ times during the stabilised addition of $A$ and $B$. This routine adds terms $2^p E_{t^*} E_{t^*}^H$, $p \leq d(T_I) - \ell$, to $S_{t^*t^*}$. Hence, the error on $t^* \times t^*$ is bounded by

$$c_{\mathrm{sp}} \sum_{\ell=0}^{d(T_I)} 2^{d(T_I)-\ell} \|E_{t^*}\|_2^2 \leq c_{\mathrm{sp}} 2^{d(T_I)+1} \|E_{t^*}\|_2^2 \leq c c_{\mathrm{sp}} |I| \varepsilon.$$

The last estimate follows from the fact that the depth $d(T_I)$ of $T_I$ scales at most like $\log_2 |I|$. Since all other blocks coincide with the blocks of $S_{\mathcal{H}}$, which have accuracy $\varepsilon$, the blockwise estimates give the global estimate

$$\|A + B - \tilde{S}_{\mathcal{H}}\|_2 \leq c c_{\mathrm{sp}}^2 d(T_{I\times I})|I|\varepsilon \leq c c_{\mathrm{sp}}^2 d(T_I)|I|\varepsilon$$

due to Lemma 2.1. □

## 4 Approximate addition of low-rank matrices

The rounded addition of two low-rank matrices is the central operation of the $\mathcal{H}$-matrix arithmetic and determines the efficiency of $\mathcal{H}$-matrices since a singular value decomposition of the sum has to be computed. The SVD of matrices $\mathbb{C}^{m\times n}$, $m \geq n$, is an expensive operation. From [10, §5.2.9 and §5.4.5] it can be seen that the cost of computing an SVD for general matrices from $\mathbb{C}^{m\times n}$ is $21mn^2$. However, for matrices $A = UV^H \in \mathbb{C}_k^{m\times n}$ it is possible to compute an SVD with complexity $\mathcal{O}(k^2(m+n))$. A method based on Gram matrices has been proposed in [14]. In [1, p. 70] the following improved variant having the complexity $6(k_A+k_B)^2(m+n) + 22(k_A+k_B)^3$ was proposed which computes the SVD of $A+B$ in $\mathbb{C}_k^{m\times n}$ for $k \leq k_A + k_B$. Assume that $QR$ decompositions $U = Q_U R_U$ and $V = Q_V R_V$ of $U := [U_A, U_B]$ and $V := [V_A, V_B]$ have been computed. Due to $UV^H = Q_U R_U R_V^H Q_V^T$, the SVD of the $m \times n$ matrix $UV^H$ simplifies to the SVD of the $k \times k$ matrix $R_U R_V^H$, where $k := k_A + k_B$. Since the rounded addition is the most time-consuming part

in the arithmetic of hierarchical matrices, improvements for this operation will directly improve the efficiency of $\mathcal{H}$-matrix operations. Therefore, it is worth investigating other algorithms for the rounded addition. If instead of the outer-product representation $A = UV^H$ of $A$ the representation by its singular value decomposition is used, the numerical effort can be further reduced. Assume that

$$A = U_A \Sigma_A V_A^H \quad \text{and} \quad B = U_B \Sigma_B V_B^H, \tag{4}$$

where $U_A, V_A, U_B$, and $V_B$ have orthonormal columns and $\Sigma_A \in \mathbb{R}^{k_A \times k_A}$ and $\Sigma_B \in \mathbb{R}^{k_B \times k_B}$ are diagonal matrices. Then

$$A + B = [U_A, U_B] \begin{bmatrix} \Sigma_A & \\ & \Sigma_B \end{bmatrix} [V_A, V_B]^H.$$

Assume that $k_A \geq k_B$. In order to reestablish a representation of type (4), we have to orthogonalise the columns of the matrices $[U_A, U_B]$ and $[V_A, V_B]$. Let $X_U := U_A^H U_B \in \mathbb{C}^{k_A \times k_B}$ and $Y_U := U_B - U_A X_U \in \mathbb{C}^{m \times k_B}$. Furthermore, let $Q_U R_U = Y_U$, $Q_U \in \mathbb{C}^{m \times k_B}$, be a $QR$ decomposition of $Y_U$. Then

$$[U_A, U_B] = \begin{bmatrix} U_A, Q_U \end{bmatrix} \begin{bmatrix} I & X_U \\ & R_U \end{bmatrix}$$

is a $QR$ decomposition of $[U_A, U_B]$. Similarly,

$$[V_A, V_B] = \begin{bmatrix} V_A, Q_V \end{bmatrix} \begin{bmatrix} I & X_V \\ & R_V \end{bmatrix}$$

is a $QR$ decomposition of $[V_A, V_B]$, where $X_V := V_A^H V_B \in \mathbb{C}^{k_A \times k_B}$ and $Q_V R_V = Y_V$ is a $QR$ decomposition of $Y_V := V_B - V_A X_V \in \mathbb{C}^{n \times k_B}$. We obtain

$$A + B = [U_A, Q_U]\, M\, [V_A, Q_V]^H,$$

where

$$M := \begin{bmatrix} \Sigma_A + X_U \Sigma_B X_V^H & X_U \Sigma_B R_V^H \\ R_U \Sigma_B X_V^H & R_U \Sigma_B R_V^H \end{bmatrix} \in \mathbb{C}^{(k_A + k_B) \times (k_A + k_B)}.$$

Using the SVD $M = \hat{U} \hat{\Sigma} \hat{V}^H$ of the (small) matrix $M$, one has that

$$A + B = ([U_A, Q_U] \hat{U})\, \hat{\Sigma}\, ([V_A, Q_V] \hat{V})^H$$

is a SVD of $A + B$, because $[U_A, Q_U]\hat{U}$ and $[V_A, Q_V]\hat{V}$ both are unitary.

For the complexity analysis we concentrate on those terms which depend on $m$. The orthogonalisation of $[U_A, U_B]$ using the previous method requires

| | |
|---|---|
| Computing $X_U$ | $2k_A k_B m$ |
| Computing $Y_U$ | $(2k_A + 1) k_B m$ |
| Decomposing $Y_U$ | $4k_B^2 m$ |
| | $(4k_A + 4k_B + 1) k_B m$ |

operations while the orthogonalisation of $[U_A, U_B]$ using the $QR$ decomposition needs $4(k_A + k_B)^2 m$ operations. If $k := k_A = k_B$, then the proposed variant requires $(8k+1)km$ operations, while $16k^2 m$ operations are needed to decompose $[U_A, U_B]$. The main advantage of this new algorithm, however, is that it is mainly based on matrix-matrix multiplications, which nowadays much attention is paid to when optimising code on a given platform. Taking into account the other parts of the algorithm we obtain the following complexity estimate.

**Theorem 4.1.** *Let $A \in \mathbb{C}_{k_A}^{m\times n}$, $B \in \mathbb{C}_{k_B}^{m\times n}$, and $k \in \mathbb{N}$ with $k \le k_A + k_B$. Then a matrix $S \in \mathbb{C}_k^{m\times n}$ satisfying*

$$\|A + B - S\| = \min_{M\in\mathbb{C}_k^{m\times n}} \|A + B - M\|$$

*with respect to any unitary invariant norm $\|\cdot\|$ can be computed with $(8k_Ak_B + 6k_B^2 + 2k_A^2 + k_B)(m + n) + 22(k_A + k_B)^3$ operations.*

In Table 1 we compare the old and the new rounded addition routines for three problem sizes. The presented CPU times are the times for 10 000 additions with accuracy $\varepsilon = 1_{10}{-}2$. Apparently, using the modified addition algorithm almost half of the time can be saved.

| $m \times n$ | $k_A$ | $k_B$ | time old | time new | gain |
|---|---|---|---|---|---|
| $200 \times 100$ | 8 | 5 | 5.23s | 4.78s | 9% |
| $300 \times 200$ | 10 | 7 | 12.57s | 8.51s | 32% |
| $400 \times 200$ | 11 | 8 | 16.05s | 9.92s | 38% |
| $600 \times 300$ | 12 | 9 | 30.05s | 15.94s | 47% |
| $800 \times 400$ | 13 | 10 | 45.97s | 23.82s | 48% |

Table 1: Old and new rounded addition.

## 5 Coarsening $\mathcal{H}$-Matrices

The previous stabilised addition is of particular importance when the accuracy of previously computed $\mathcal{H}$-matrix approximants is to be significantly reduced for the purpose of preconditioning. In this situation it is not only useful to remove further singular triplets from the SVD of each block, the block structure itself can be improved by unifying neighbouring blocks. Fig. 1 shows an $\mathcal{H}$-matrix before and after coarsening. Unifying blocks may even allow to increase the rank while decreasing the amount of storage. In [12] it was demonstrated that improving the partition leads to a significant reduction of the $\mathcal{H}$-matrix arithmetic.

In this section we describe how a given matrix $A \in \mathcal{H}(T_{I\times J}, k)$ is approximated by a matrix $A' \in \mathcal{H}(T'_{I\times J}, k')$, where $T'_{I\times J}$ is a sub-tree of $T_{I\times J}$ with the same root $I \times J$. While in [13] the coarsening of $\mathcal{H}$-matrices with fixed rank $k$ has been investigated for the purpose of multiplying $\mathcal{H}$-matrices, we want $A'$ to approximate $A$ with given accuracy $\varepsilon > 0$. In this case, the required blockwise rank $k'$ would significantly increase if $T_{I\times J}$ is coarsened to the root $I \times J$. Hence, the coarsened tree $T'_{I\times J}$ has to be found such that the cost of the approximant $A'$ does not increase compared with $A$, while the computation of $A'$ has almost linear complexity.

Assume that the matrix $A_d \in \mathcal{H}(T_d, k_\varepsilon)$, $k_\varepsilon \le k$, has been generated from $A$ by approximating each block $A_b$, $b \in \mathcal{L}(T_{I\times J})$, with accuracy $\varepsilon$ using the technique from Sect. 4. Here, $T_d := T_{I\times J}$ is the initial tree of depth $d = d(T_{I\times J})$. Since $A$ is approximated on each block with accuracy $\varepsilon$, for the Frobenius norm it holds that

$$\|A - A_d\|_F \le \varepsilon\|A\|_F. \tag{5}$$

This first coarsening step requires

$$\sum_{t\times s\in\mathcal{L}(T_{I\times J})} 16k^2(|t| + |s|) + 22k^3 \le 16c_{\mathrm{sp}}k^2[d(T_I)|I| + d(T_J)|J|] + 22c_{\mathrm{sp}}k^3[|I| + |J|]/n_{\min}$$

arithmetical operations due to (1a,b).

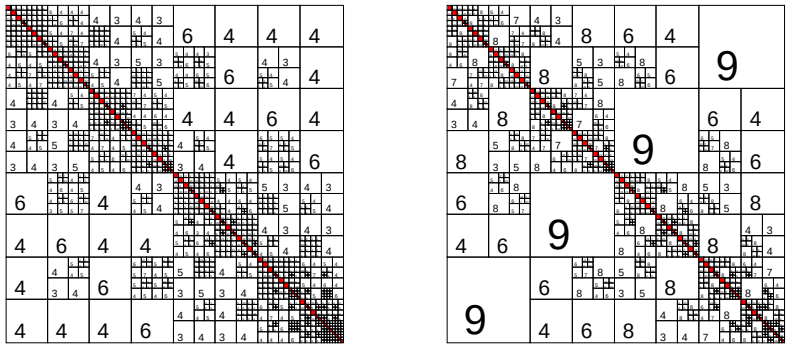

Figure 1: $\mathcal{H}$-matrix before and after coarsening (the numbers indicate the local rank).

In a second step we improve the block structure. For this purpose consider a $2 \times 2$ block matrix

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \approx U \Sigma V^H$$

consisting of four low-rank matrices $A_i = U_i \Sigma_i V_i^H$, $i = 1, \ldots, 4$, each having rank at most $k$. Assume this matrix is to be approximated by a single matrix $U \Sigma V^H \in \mathbb{C}_{k_\varepsilon}^{m \times n}$ with $k_\varepsilon \in \mathbb{N}$. Since

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} = \begin{bmatrix} A_1 & \\ & \end{bmatrix} + \begin{bmatrix} & A_2 \\ & \end{bmatrix} + \begin{bmatrix} & \\ A_3 & \end{bmatrix} + \begin{bmatrix} & \\ & A_4 \end{bmatrix},$$

this problem may be regarded as a rounded addition of four low-rank matrices. Therefore, an approximation with accuracy $\varepsilon$ can be computed in $\mathbb{C}_{k_\varepsilon}^{m \times n}$ using the SVD of low-rank matrices. Note that $k_\varepsilon$ may be larger than $k$; in general $k_\varepsilon$ is bounded only by $4k$.

Compared with the rounded addition of general low-rank matrices, the presence of zeros should be taken into account. Since

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} = \hat{U} \hat{\Sigma} \hat{V}^H,$$

where

$$\hat{U} := \begin{bmatrix} U_1 & U_2 & & \\ & & U_3 & U_4 \end{bmatrix}, \quad \hat{V} := \begin{bmatrix} V_1 & & V_3 & \\ & V_2 & & V_4 \end{bmatrix}, \quad \text{and} \quad \hat{\Sigma} = \begin{bmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \Sigma_3 & \\ & & & \Sigma_4 \end{bmatrix},$$

it is enough to orthonormalise $[U_1, U_2]$, $[U_3, U_4]$, $[V_1, V_3]$, and $[V_2, V_4]$. The number of arithmetical operations can be estimated as

| | |
|---|---|
| Computing the $QR$ decompositions | $2(8k+1)k(m+n)$ |
| Building the unitary factors | $2(2k)^2(m+n)$ |
| | $\sim\ 24k^2(m+n)$ |

The amount of operations can be reduced if each of the matrices $[A_1, A_2]$ and $[A_3, A_4]$ are agglomerated before agglomerating the results. However, in this case we cannot expect to obtain a best approximation.

Using the previous agglomeration of a $2 \times 2$ block matrix, the following recursion defines a sequence of block cluster trees $T_\ell$ and an associated sequence of approximants $A_\ell \in \mathcal{H}(T_\ell, k_\varepsilon)$, $\ell = d-1, \ldots, 1$.

Let $T_\ell$ arise from $T_{\ell+1}$ by removing the sons of each block $b \in T_{\ell+1}^{(\ell)} \setminus \mathcal{L}(T_{\ell+1})$ in the $\ell$-th level of $T_{\ell+1}$. $A_\ell$ results from $A_{\ell+1}$ by the previous agglomeration procedure applied to such blocks $b$. Note that in order to guarantee that a possible positivity of $A_{\ell+1}$ is preserved for $A_\ell$, we have to employ the stabilisation technique from Sect. 3.

Since we want to avoid that $A_\ell$ requires more storage than $A_{\ell+1}$, we stop the agglomeration process in blocks for which the required rank $k_\varepsilon$ is such that the agglomeration is not worthwhile. Assume that the sub-blocks $S(b^*)$ of a block $b^* = t^* \times s^* \in T_{\ell+1}^{(\ell)} \setminus \mathcal{L}(T_{\ell+1})$ in $A_{\ell+1}$ are low-rank matrices with ranks

$$k_{t\times s} \quad \text{for} \quad t \times s \in S(b^*). \tag{6}$$

In order to agglomerate a non-admissible block, it is first converted to the outer product representation using the SVD. By comparing the original storage costs of $(A_{\ell+1})_{b^*}$ and the costs of $(A_\ell)_{b^*}$, it is easy to check whether the coarsening leads to an increment of the cost of the approximant. If

$$k_{t^*\times s^*}(|t^*| + |s^*|) \le \sum_{t\times s\in S(b^*)} k_{t\times s}(|t| + |s|), \tag{7}$$

then the block cluster tree $T_{\ell+1}$ is modified by replacing the sons $S(b^*)$ of $b^*$ by the new leaf $b^*$. If this condition is not satisfied, then the sons of $b^*$ will be kept in the block cluster tree. This procedure can then be applied to the leaves of the new block cluster tree until (7) is not satisfied.

It is obvious that the previous recursion cannot increase the amount of storage. What remains is to estimate the accuracy of $A'$ and the computational cost of its computation. Since each block has accuracy $\varepsilon > 0$, this property is inherited by the whole matrix with respect to the Frobenius norm, i.e,

$$\|A_{\ell+1} - A_\ell\|_F \le \varepsilon \|A_{\ell+1}\|_F. \tag{8}$$

The following lemma describes the accuracy of $A'$ compared with the accuracy of $A$.

**Lemma 5.1.** *Let $A \in \mathcal{H}(T_{I\times J}, k)$. Then there is $c > 0$ such that $\|A - A'\|_F \le c\, d(T_{I\times J})\varepsilon\|A\|_F$.*

*Proof.* From (8) and $\|A_\ell\|_F \le \|A_{\ell+1}\|_F + \|A_{\ell+1} - A_\ell\|_F \le (1+\varepsilon)\|A_{\ell+1}\|_F$ we have that

$$\begin{aligned}\|A_d - A_1\|_F = \|\sum_{\ell=1}^{d-1}(A_{\ell+1} - A_\ell)\|_F &\le \sum_{\ell=1}^{d-1}\|A_{\ell+1} - A_\ell\|_F \le \varepsilon\sum_{\ell=1}^{d-1}\|A_{\ell+1}\|_F \\ &\le \varepsilon\sum_{\ell=1}^{d-1}(1+\varepsilon)^{d-\ell-1}\|A_d\|_F \le [(1+\varepsilon)^{d-1} - 1]\|A_d\|_F.\end{aligned}$$

Using (5), we obtain that

$$\begin{aligned}\|A - A_1\|_F &\le \|A - A_d\|_F + \|A_d - A_1\|_F \le \varepsilon\|A\|_F + [(1+\varepsilon)^{d-1} - 1]\|A_d\|_F \\ &\le \varepsilon\|A\|_F + [(1+\varepsilon)^{d-1} - 1](\|A - A_d\|_F + \|A\|_F) \\ &\le \left\{\varepsilon + (1+\varepsilon)[(1+\varepsilon)^{d-1} - 1]\right\}\|A\|_F = [(1+\varepsilon)^d - 1]\|A\|_F.\end{aligned}$$

The assertion follows from $(1+\varepsilon)^d - 1 \sim d\varepsilon$ as $\varepsilon \to 0$. □

The number of arithmetical operations of the above construction is determined by the number of operations required for the SVD of each non-admissible block and the numerical effort for the agglomeration of each $b \in T_{I\times J} \setminus \mathcal{L}(T_{I\times J})$. The computational cost is estimated in the following lemma. For this purpose we consider a single block $b \in T_{I\times J}$ in which the described agglomeration stops due to the violation of (7) by the father block of $b$. Without loss of generality we identify $b$ with $I \times J$ and assume that (7) holds for all $t \times s \in T_{I\times J} \setminus \mathcal{L}(T_{I\times J})$.

**Lemma 5.2.** *Assume that* (7) *holds in each step of the above coarsening procedure applied to* $T_{I\times J}$ *such that* $T_{I\times J}$ *is coarsened to* $I \times J$. *Let* $A \in \mathcal{H}(T_{I\times J}, k)$. *The storage requirements of* $A'$ *are bounded by those of* $A$ *while the resulting blockwise rank of* $A'$ *is bounded by* $c_{\mathrm{sp}} k d(T_I)$. *The required computational cost is of the order*

$$c_{\mathrm{sp}}^3 k^2 d^3(T_{I\times J})[|I| + |J|].$$

*Proof.* Let $k_{t\times s}$ be as in (6). Similarly to the case of a blockwise constant rank, the costs of coarsening $T_{I\times J}$ can be estimated as

$$\sum_{t\times s\in T_{I\times J}} k_{t\times s}^2(|t| + |s|),$$

where we have omitted terms which do depend neither on $|t|$ nor $|s|$. Using (7), the cost of each block $t^* \times s^* \in T_{I\times J}$ can be estimated by the sum over its leaves:

$$k_{t^*\times s^*}(|t^*| + |s^*|) \le \sum_{t\times s\in\mathcal{L}(T_{t^*\times s^*})} k_{t\times s}(|t| + |s|) \le k \sum_{t\times s\in\mathcal{L}(T_{t^*\times s^*})} |t| + |s|.$$

From (1a,b) it follows that $k_{t^*\times s^*} \le c_{\mathrm{sp}} d(T_{t^*\times s^*})k$. With the last estimate we obtain

$$\sum_{t\times s\in T_{I\times J}} k_{t\times s}^2(|t| + |s|) \le c_{\mathrm{sp}}^2 d^2(T_{I\times J})k^2 \sum_{t\times s\in T_{I\times J}} |t| + |s| \le c_{\mathrm{sp}}^3 d^3(T_{I\times J})k^2[|I| + |J|]$$

due to another application of (1a,b). □

## 6 Numerical experiments

For the numerical tests we consider the mixed boundary value problem

$$-\Delta u = f \quad \text{in } \Omega, \tag{9a}$$
$$u = g_D \quad \text{on } \Gamma_D, \tag{9b}$$
$$\partial_n u = 0 \quad \text{on } \Gamma_N, \tag{9c}$$

where $\Omega$ is the coil domain[1] shown in Fig. 2. The domain $\Omega$ is connected to other devices on its left and right end, where we impose Dirichlet boundary conditions. The respective part of the boundary will be denoted by $\Gamma_D$. On the larger part $\Gamma_N := \Gamma \setminus \Gamma_D$ of the boundary $\Gamma := \partial\Omega$ we impose Neumann boundary conditions.

This system is to be solved for the potential $u$, which can be calculated by making use of Green's representation formula (cf. [17]) in the interior of the conductor

$$u(x) = \mathcal{V}[\partial_n u](x) - \mathcal{K}[u](x), \quad x \in \Omega, \tag{10}$$

where $\mathcal{V}$ represents the *single-layer potential operator*

$$(\mathcal{V}v)(x) := \frac{1}{4\pi}\int_\Gamma \frac{v(y)}{|x-y|}\,\mathrm{d}s_y, \quad x \in \mathbb{R}^3 \setminus \Gamma,$$

acting on $v$ on the boundary $\Gamma$. $\mathcal{K}$ is the *double-layer potential operator*

$$(\mathcal{K}v)(x) := \frac{1}{4\pi}\int_\Gamma v(y)\,\partial_{n_y}\frac{1}{|x-y|}\,\mathrm{d}s_y, \quad x \in \mathbb{R}^3 \setminus \Gamma.$$

[1] The computational geometry is by courtesy of ABB Schweiz AG

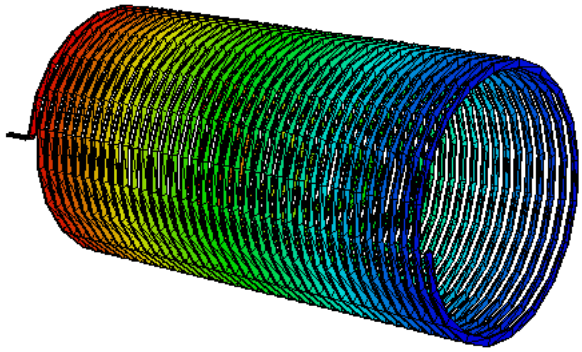

Figure 2: A domain with dominating Neumann boundary.

Applying the trace operators $\gamma_0$ and $\gamma_1$ to the representation formula (10) leads to the following boundary integral equations

$$\begin{bmatrix}\gamma_0 u\\ \gamma_1 u\end{bmatrix} = \begin{bmatrix}\frac{1}{2}I - \mathcal{K} & \mathcal{V}\\ \mathcal{D} & \frac{1}{2}I + K'\end{bmatrix}\begin{bmatrix}\gamma_0 u\\ \gamma_1 u\end{bmatrix}, \quad x \in \Gamma, \tag{11}$$

in which

$$(\mathcal{K}'v)(x) := \frac{1}{4\pi}\int_\Gamma v(y)\,\partial_{n_x}\frac{1}{|x-y|}\,\mathrm{d}s_y, \quad x \in \mathbb{R}^3 \setminus \Gamma$$

denotes the adjoint of $\mathcal{K}$ and $\mathcal{D}$ is the *hypersingular operator* which results from applying the negative Neumann-trace to the double-layer potential operator. Let $\tilde{g}_D$ and $\tilde{g}_N$ denote the canonical extensions of $g_D$ and $g_N$ to $\Gamma$. Setting $\tilde{u} := u - \tilde{g}_D$ and $\tilde{t} := t - \tilde{g}_N$, we have to compute $\tilde{u} \in \tilde{H}^{1/2}(\Gamma_N)$ and $\tilde{t} \in \tilde{H}^{-1/2}(\Gamma_D)$. Here, $H^s(\Gamma_0)$ and $\tilde{H}^s(\Gamma_0)$, $s \geq 0$, denote Sobolev spaces on a part $\Gamma_0 \subset \Gamma$ of the boundary $\Gamma$:

$$H^s(\Gamma_0) := \{u|_{\Gamma_0} : u \in H^s(\Gamma)\} \quad \text{and} \quad \tilde{H}^s(\Gamma_0) := \{u \in H^s(\Gamma_0) : \operatorname{supp} u \subset \overline{\Gamma}_0\}$$

with the norm

$$\|u\|_{H^s(\Gamma_0)} := \inf\{\|u|_{\Gamma_0}\|_{H^s(\Gamma)} : u \in H^s(\Gamma)\}.$$

The negative Sobolev space $H^{-s}(\Gamma_0)$ on $\Gamma_0$ is defined as the dual of $H^s(\Gamma_0)$.

Since $\tilde{u} = 0$ on $\Gamma_D$ and $\tilde{t} = 0$ on $\Gamma_N$, we obtain from (11)

$$-\mathcal{V}\tilde{t} + \mathcal{K}\tilde{u} = \mathcal{V}\tilde{g}_N - \left(\frac{1}{2}\mathcal{I} + \mathcal{K}\right)\tilde{g}_D \quad \text{on } \Gamma_D, \tag{12a}$$

$$\mathcal{K}'\tilde{t} + \mathcal{D}\tilde{u} = \left(\frac{1}{2}\mathcal{I} - \mathcal{K}'\right)\tilde{g}_N - \mathcal{D}\tilde{g}_D \quad \text{on } \Gamma_N. \tag{12b}$$

Equation (10) together with (11) make up the so-called *symmetric* boundary integral formulation of the mixed boundary value problem (9). It is known (see for instance [17]) that $\mathcal{V} : \tilde{H}^{-1/2}(\Gamma_D) \to H^{1/2}(\Gamma_D)$ is continuous and $\tilde{H}^{-1/2}(\Gamma_D)$-elliptic, i.e.,

$$\langle \mathcal{V}v, v\rangle_{L^2(\Gamma_D)} \geq c_{\mathcal{V}}\|v\|^2_{\tilde{H}^{-1/2}(\Gamma_D)} \quad \text{for all } v \in \tilde{H}^{-1/2}(\Gamma_D)$$

and that $\mathcal{D} : \tilde{H}^{1/2}(\Gamma_N) \to H^{-1/2}(\Gamma_N)$ is continuous and $\tilde{H}^{1/2}(\Gamma_N)$-elliptic, i.e.,

$$\langle \mathcal{D}v, v\rangle_{L^2(\Gamma_N)} \geq c_{\mathcal{D}}\|v\|^2_{\tilde{H}^{1/2}(\Gamma_D)} \quad \text{for all } v \in \tilde{H}^{1/2}(\Gamma_D)$$

provided $\Gamma_D$ has a positive measure. Hence, (12a,b) is uniquely solvable since the Schur complement $\mathcal{D} + \mathcal{K}'\mathcal{V}^{-1}\mathcal{K} : \tilde{H}^{1/2}(\Gamma_N) \to H^{-1/2}(\Gamma_N)$ is continuous and $\tilde{H}^{1/2}(\Gamma_N)$-elliptic. If $\Gamma_D = \emptyset$ such that $\Gamma_N = \partial\Omega$, then $\mathcal{D}$ will not be invertible since (9) becomes a pure Neumann problem.

After discretising (12a,b) by a Galerkin method, we obtain a linear system with the partially known Neumann data $t_h := \sum_{i=1}^{n'} t_i\psi_i$, where $\psi_i$ are piecewise constants such that $W_h := \operatorname{span}\{\psi_i : i = 1, \ldots, n'\} \subset H^{-1/2}(\Gamma)$. Furthermore, let $V_h := \operatorname{span}\{\varphi_j : j = 1, \ldots, n\}$ be made of piecewise linears. The discrete variational formulation of (12a,b) leads to the following algebraic system of equations for the unknown coefficients $u \in \mathbb{R}^{n_D}$ and $t \in \mathbb{R}^{n'_N}$ of $u_h$ and $t_h$

$$\begin{bmatrix} -V & K \\ K^T & D \end{bmatrix} \begin{bmatrix} t \\ u \end{bmatrix} = \begin{bmatrix} V & -\frac{1}{2}M - K \\ \frac{1}{2}M - K^T & -D \end{bmatrix} \begin{bmatrix} \tilde{g}_N \\ \tilde{g}_D \end{bmatrix} =: \begin{bmatrix} f_N \\ f_D \end{bmatrix}.$$

The entries of the above matrices are

$$V_{k\ell} = (\mathcal{V}\psi_\ell, \psi_k)_{L^2}, \quad K_{kj} = (\mathcal{K}\varphi_j, \psi_k)_{L^2}, \quad D_{ij} = (\mathcal{D}\varphi_j, \varphi_i)_{L^2},$$

where $k, \ell = 1, \ldots, n'_N$ and $i, j = 1, \ldots, n_D$. Hence, for boundary value problems having a small Dirichlet part, we can expect numerical difficulties since the smallest eigenvalue of the discrete of $\mathcal{D}$ decreases with the size of $\Gamma_D$.

Since for the domain from Fig. 2, $\Gamma_D$ is tiny compared with the Neumann boundary $\Gamma_N$, the above problem is ill-conditioned and preconditioning is required. Using the $\mathcal{H}$-arithmetic, one can construct a block $LU$ decomposition, in which coarse approximations of Cholesky factors of $V$ and $D$ appear. Before computing the Cholesky decomposition, it is helpful to reduce the complexity by the agglomeration technique from Sect. 5. The introduced error leads to a perturbation of the spectrum such that, depending on the accuracy, $D$ may become indefinite.

For preconditioning $A$ we could obviously approximate the following $LU$ decomposition

$$A = \begin{bmatrix} L_1 & \\ -X^T & L'_2 \end{bmatrix} \begin{bmatrix} -L_1^T & X \\ & L_2'^T \end{bmatrix},$$

where $V = L_1 L_1^T$ is the Cholesky decomposition of $V$, $X$ is defined by $L_1 X = K$, and $L'_2 L_2'^T = D + X^T X$ is the Cholesky decomposition of the Schur complement. The computation of $D + X^T X$ can however be time-consuming even if $\mathcal{H}$-matrices are employed. Therefore, we use the following matrix $C$ for symmetrically preconditioning $A$

$$C := \hat{U}^{-1} \begin{bmatrix} L_1^{-T} & \\ & L_2^{-T} \end{bmatrix} \quad \text{with} \quad \hat{U} := \begin{bmatrix} I & -L_1^{-T}X \\ & I \end{bmatrix}$$

and $L_1$ and $L_2$ denote lower triangular $\mathcal{H}$-matrices such that

$$\|I - (L_1 L_1^T)^{-1} V\|_2 < \delta, \quad \|I - (L_2 L_2^T)^{-1} D\|_2 < \delta,$$

and $X$ is an $\mathcal{H}$-matrix satisfying $\|K - L_1 X\|_2 < \delta$. The computation of the Cholesky factors $L_1$ and $L_2$ with almost linear complexity is explained in [4]. Note that $L_2$ is defined to be the approximate Cholesky factor of $D$ but not of $D + X^T X$; i.e., instead of approximating the original coefficient matrix $A$, $C^T C$ approximates the matrix

$$\begin{bmatrix} -V & K \\ K^T & D - K^T V^{-1} K \end{bmatrix}^{-1}.$$

After generating the approximant using the ACA algorithm (see [7, 5]) with accuracy $\varepsilon = 1_{10}{-}6$, we recompress a copy of the coefficient matrix to a blockwise relative accuracy $\delta$ using the coarsening

| | $n = 3\,128$ | | | | | $n = 12\,520$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | precond. | | solution | | | precond. | | solution | |
| $\delta$ | time | MB | #It | time | $\delta$ | time | MB | #It | time |
| $1_{10}-3$ | 2.1s | 9.3 | 55 | 1.4s | $2_{10}-4$ | 18.5s | 36.0 | 50 | 4.8s |
| $2_{10}-3$ | 1.8s | 8.5 | 69 | 1.7s | $5_{10}-4$ | 16.4s | 32.8 | 71 | 7.2s |
| $5_{10}-3$ | 1.5s | 7.5 | 84 | 2.1s | $1_{10}-3$ | 14.1s | 30.1 | 96 | 8.9s |

Table 2: Preconditioned MinRes for the domain from Fig. 2.

procedure from Sect. 5. The hierarchical Cholesky decomposition fails to compute unless a rounding precision is used that is significantly higher than the values of $\delta$. In order to circumvent this problem, we use a stabilised variant which is based on the stabilisation technique from Sect. 3. The additional time for computing the preconditioner and the total time for the iterative solution using MinRes in Table 2 can be neglected compared with the construction of the $\mathcal{H}$-matrix approximant. Iterating without any preconditioner does not converge at all. Note that preconditioners which guarantee an asymptotic boundedness of the condition number with respect to $n$ based on the mapping properties of the operator (see [19]) will not be enough in order to obtain a reasonable convergence behaviour. The bad condition number is caused by the geometry and not by $n$.

## References


[1] M. Bebendorf. *Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen*. PhD thesis, Universität Saarbrücken, 2000. dissertation.de, Verlag im Internet, 2001. ISBN 3-89825-183-7.

[2] M. Bebendorf. Efficient inversion of Galerkin matrices of general second-order elliptic differential operators with nonsmooth coefficients. *Mathematics of Computation*, 74:1179–1199, 2005.

[3] M. Bebendorf. Hierarchical *LU* decomposition based preconditioners for BEM. *Computing*, 74:225–247, 2005.

[4] M. Bebendorf. Why approximate *LU* decompositions of finite element discretizations of elliptic operators can be computed with almost linear complexity. Preprint 8, Max-Planck-Institute MiS, Leipzig, 2005. submitted.

[5] M. Bebendorf and R. Grzibowski. Accelerating Galerkin BEM for Linear Elasticity using Adaptive Cross Approximation. *Mathematical Methods in the Applied Sciences*, 29:1721–1747, 2006.

[6] M. Bebendorf and W. Hackbusch. Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with $L^\infty$-coefficients. *Numerische Mathematik*, 95(1):1–28, 2003.

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

[8] G. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrica*, 1:211–218, 1936.

[9] K. Giebermann. Multilevel approximation of boundary integral operators. *Computing*, 67:183–207, 2001.

[10] G. H. Golub and Ch. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[11] L. Grasedyck. *Theorie und Anwendungen Hierarchischer Matrizen*. PhD thesis, Universität Kiel, 2001.

[12] L. Grasedyck. Adaptive recompression of $\mathcal{H}$-matrices for BEM. *Computing*, 74:205–223, 2005.

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

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

[15] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic: general complexity estimates. *Journal of Computational and Applied Mathematics*, 125(1-2):479–501, 2000. Numerical analysis 2000, Vol. VI, Ordinary differential equations and integral equations.

[16] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

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

[18] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quarterly Journal of Mathematics. Oxford 2 Series*, 11:50–59, 1960.

[19] O. Steinbach and W. Wendland. The construction of some efficient preconditioners in the boundary element method. *Advances in Computational Mathematics*, 9:191–216, 1998.