Computing 70, 295–334 (2003)
Digital Object Identifier (DOI) 10.1007/s00607-003-0019-1

Computing
Printed in Austria

Construction and Arithmetics of $\mathscr{H}$-Matrices

Lars Grasedyck and **Wolfgang Hackbusch**, Leipzig, Germany

Received November 28, 2002; revised April 15, 2003
Published online: July 28, 2003

Abstract

In previous papers hierarchical matrices were introduced which are data-sparse and allow an approximate matrix arithmetic of nearly optimal complexity. In this paper we analyse the complexity (storage, addition, multiplication and inversion) of the hierarchical matrix arithmetics. Two criteria, the sparsity and idempotency, are sufficient to give the desired bounds. For standard finite element and boundary element applications we present a construction of the hierarchical matrix format for which we can give explicit bounds for the sparsity and idempotency.

AMS Subject Classification: 65F05, 65F30, 65F50.

Keywords: Hierarchical matrices, data-sparse approximations, formatted matrix operations, fast solvers.

1 Introduction

1.1 Overview

In [8] a new format for the representation of matrices was introduced, the so-called hierarchical matrices or shortly $\mathscr{H}$-matrices. This format is well-suited for the data-sparse representation of matrices arising in the boundary element method or for the approximation of the inverse to a finite element discretisation of an elliptic partial differential operator. In subsequent papers, several model problems were analysed and for each of them a suitable $\mathscr{H}$-matrix format was defined. A short overview and an introduction to hierarchical matrices can be found in [3].

In this paper we do not describe the various applications of the $\mathscr{H}$-matrix arithmetic, but present a precise complexity analysis. It turns out that such an analysis can be based on two criteria, namely the sparsity and idempotency of the underlying tree. Corresponding to the exact matrix operations $+, \cdot$ we define the so-called *formatted* matrix operations $\oplus$, $\odot$ that allow us to compute an approximate inverse to an $\mathscr{H}$-matrix in almost linear complexity. For standard finite element and boundary element applications we are able to give a construction of the $\mathscr{H}$-matrix format where we can give explicit bounds for the sparsity and idempotency.

The rest of the paper is organised as follows. The next subsections give a short introduction to $\mathcal{H}$-matrices. In Section 2 we present the algorithms for the formatted arithmetic operations within the set of $\mathcal{H}$-matrices and estimate their complexity. Section 3 describes the image of the inversion operator in the set of $\mathcal{H}$-matrices and introduces the admissibility condition that allows us to approximate efficiently (BEM) stiffness matrices or the inverse to a (FEM) stiffness matrix in the set of $\mathcal{H}$-matrices. Based upon the admissibility condition we construct the hierarchical structures and $\mathcal{H}$-matrices in Section 4. The theoretical results are confirmed by numerical tests which are presented in Section 5.

1.2 $R(k)$-Matrices

The basic building blocks for $\mathcal{H}$-matrices are matrices of low rank (as compared to their size). We use a data sparse representation for this kind of matrices.

Definition 1.1 ($R(k)$-matrix representation) *Let $k, n, m \in \mathbb{N}_0$. A matrix $M \in \mathbb{R}^{n\times m}$ is called an $R(k)$-matrix (given in $R(k)$-representation) if M is given in factorised form*

$$M = AB^T, \qquad A \in \mathbb{R}^{n\times k},\ B \in \mathbb{R}^{m\times k}, \tag{1.1}$$

with A, B in full matrix representation.

Throughout this paper the storage is measured by the number of floating point numbers to be stored, while the cost of an operation is given by the number of elementary operations $+, -, \cdot, /$.

Remark 1.2 (storage and matrix-vector product) *The storage requirements $N_{F,St}(n,m)$ for a matrix $M \in \mathbb{R}^{n\times m}$ in full matrix representation is $N_{F,St}(n,m) = nm$. The storage requirements $N_{R,St}(n,m,k)$ for an $n\times m$ $R(k)$-matrix M is*

$$N_{R,St}(n,m,k) = k(n+m). \tag{1.2}$$

The complexity $N_{F\cdot v}(n,m)$ and $N_{R\cdot v}(n,m,k)$ for the computation of the matrix-vector product of M in full matrix and $R(k)$-matrix representation is

$$N_{F\cdot v}(n,m) = 2nm - n, \quad N_{R\cdot v}(n,m,k) = 2k(n+m) - n - k.$$

In the next lemma, the term ‘truncated’ will appear in two meanings. First in part (a) the truncated singular value decomposition (SVD) and the truncated QR-decomposition are the exact ones, where the corresponding factors are reduced to the non-zero part. In part (b) of the lemma, truncation from rank k to $k' < k$ includes loss of information.

Lemma 1.3 (truncated SVD, truncation) (*a*) *Let $R = AB^T \in \mathbb{R}^{n\times m}$ be an $R(k)$-matrix. A truncated singular value decomposition of R can be computed with complexity $N_{R,\mathrm{SVD}}(n,m,k) \le 5k^2(n+m) + 23k^3$ as follows:*

1. *Calculate a truncated QR-decomposition* $A = Q_A R_A$ *of* A, $Q_A \in \mathbb{R}^{n\times k}, R_A \in \mathbb{R}^{k\times k}$.
2. *Calculate a truncated QR-decomposition* $B = Q_B R_B$ *of* B, $Q_B \in \mathbb{R}^{m\times k}, R_B \in \mathbb{R}^{k\times k}$.
3. *Calculate a singular value decomposition* $R_A R_B^T = \tilde{U}\Sigma\tilde{V}^T$ *of* $R_A R_B^T$.
4. *Define* $U := Q_A\tilde{U} \in \mathbb{R}^{n\times k}$ *and* $V := Q_B\tilde{V} \in \mathbb{R}^{m\times k}$.

Then $R = U\Sigma V^T$ *is a (truncated) SVD. Due to* [5, *Sections* 5.2.9 *and* 5.4.5], *the complexity of the previous steps is*

QR − decomposition of A:	$4nk^2$	
QR − decomposition of B:	$4mk^2$	
multiplication of $R_A R_B^T$:		$2k^3$
SVD of $R_A R_B^T$:		$21k^3$
Multiplication of $Q_A\widetilde{U}$ and $Q_B\widetilde{V}$:	$2nk^2 + 2mk^2$	
Altogether: $N_{R,\mathrm{SVD}}(n,m,k) =$	$6k^2(n+m)+$	$23k^3$

(*b*) *A truncation of an* $R(k)$*-matrix* R *to rank* $k' \le k$ *is defined as the best approximation with respect to the Frobenius and spectral norm of* R *in the set of* $R(k')$*-matrices. This can be computed by using the first* k' *columns of the matrices* $U\Sigma$ *and* V *from the truncated singular value decomposition of* R *with the same complexity as above. We denote the truncation from rank* k *to* k' *by the symbol*

$$\mathscr{T}^R_{k'\leftarrow k}. \tag{1.3}$$

If $k' \ge k$, $\mathscr{T}^R_{k'\leftarrow k}$ *is the identity. In the representation* (1.1), *the matrices* A, B *are extended by* $k' - k$ *zero columns.*

We remark that the truncation in part (b) becomes non-unique when the k'th and $(k'+1)$st singular values are equal. Consequently, all operators defined below and involving a truncation may be non-unique.

Definition 1.4 (formatted addition) *The formatted addition* $R \oplus S$ *of two* $n \times m$ $R(k)$*-matrices* R *and* S *is defined as a truncation of* $R + S$ *to the set of* $R(k)$*-matrices, i.e.,* $R \oplus S := \mathscr{T}^R_{k\leftarrow 2k}(R+S)$.

Note that $\oplus$ is commutative, but in general not distributive (i.e., $(A \oplus B) \oplus C$ and $A \oplus (B \oplus C)$ may differ).

Remark 1.5 *The formatted addition can be computed with complexity* $N_{R,\oplus}(n,m,k) \le 24k^2(n+m) + 184k^3$.

Proof. Use the truncation of Lemma 1.3b for the $R(2k)$-matrix $R + S$. □

Lemma 1.6 (spectral and Frobenius norm) *The spectral and Frobenius norm of an* $n \times m$ $R(k)$*-matrix* R *can be computed as in Lemma* 1.3*a with complexity* $N_{R,\|\cdot\|}(n,m,k) \le 4k^2(n+m) + 23k^3$.

Proof. The norms can be obtained from the singular values, i.e., steps 1–3 from Lemma 1.3a are to be performed. □

1.3 $\mathscr{H}$-Matrices

In essence, the hierarchical structure of $\mathscr{H}$-matrices is the tree structure defined below.

1.3.1 $\mathscr{H}$-Trees T_I

Here, we give only the definition of an $\mathscr{H}$-tree and introduce some notations. The concrete construction of the tree will be discussed in §4.1.

Definition 1.7 ($\mathscr{H}$-tree, sons) *Let I be a finite set and let $T_I = (V, E)$ be a tree with vertex set V and edge set E. For a vertex $v \in V$ we define the set of sons of v as $S(v) := \{w \in V \mid (v,w) \in E\}$. The tree T_I is called an $\mathscr{H}$-tree of I, if the following conditions hold*:

$$I \text{ is the root of } T_I \text{ and } \emptyset \neq v \subset I \text{ for all } v \in V, \tag{1.4a}$$

$$\forall v \in V: \text{ either } S(v) = \emptyset \textit{ or } v = \dot{\bigcup}_{w \in S(v)} w. \tag{1.4b}$$

In (1.4b) we use the notation $\dot{\cup}$ for the disjoint union.

In the following we identify V and T_I, i.e., we write $v \in T_I$ instead of $v \in V$. The edge set E is not needed, since $S(\cdot)$ contains all information about the edges.

Definition 1.8 (descendant, father, leaf, level, depth) *Let T_I be an $\mathscr{H}$-tree. We define the descendants of a vertex $v \in T_I$ by $S^*(v) := \{w \in T_I \mid w \subset v\}$ and the uniquely determined predecessor (father) of a non-root vertex $v \in T_I$ is denoted by $\mathscr{F}(v)$. The set of leaves of the tree T_I is $\mathscr{L}(T_I) = \{v \in T_I \mid S(v) = \emptyset\}$. The levels of the tree T_I are defined as*

$$T_I^{(0)} := \{I\}, \quad T_I^{(\ell)} := \{v \in T_I \mid \mathscr{F}(v) \in T_I^{(\ell-1)}\} \qquad \text{for } \ell \in \mathbb{N},$$

and we write $\operatorname{level}(v) = \ell$ *if* $v \in T_I^{(\ell)}$. *The depth of T is defined as* $\operatorname{depth}(T) := \max\{\ell \in \mathbb{N}_0 \mid T_I^{(\ell)} \neq \emptyset\}$. *The leaves of T on level ℓ are denoted by* $\mathscr{L}(T_I, \ell) := \mathscr{L}(T_I) \cap T_I^{(\ell)}$.

The introduced notation requires implicitly $\#S(v) \neq 1$, as discussed in

Remark 1.9 (general $\mathscr{H}$-trees) *In the definition of an $\mathscr{H}$-tree the vertices were labelled by subsets of the index set I. Therefore, it is not possible that a vertex v has exactly one son w ((1.4b) would demand $v = w$). This could be overcome by denoting the vertices of an $\mathscr{H}$-tree by a tuple (v, ℓ), where $v \subset I$ and ℓ is the level number of the vertex. Then the vertex (v, ℓ) is allowed to have exactly one son $(v, \ell + 1)$. In the* *rare cases where it becomes important, we will explicitly note the level number, e.g., by* $v \in T^{(\ell)}$, *but omit the tuple notation otherwise.*

Remark 1.10 (*a*) *Any* $\mathscr{H}$*-tree* T_I *with root* I *has the property* $\dot\cup_{v\in\mathscr{L}(T_I)} v = I$, *i.e., the leaves of an* $\mathscr{H}$*-tree yield a partitioning for the index set* I.

(*b*) *For any* $\mathscr{H}$*-tree* T_I *and* $\ell \in \{0,\ldots,\operatorname{depth}(T)\}$ *there holds*

$$I = \left(\dot{\bigcup_{v\in T_I^{(\ell)}}} v\right) \dot\cup \left(\dot{\bigcup_{v\in\mathscr{L}(T_I,\ell-1)}} v\right) \dot\cup \cdots \dot\cup \left(\dot{\bigcup_{v\in\mathscr{L}(T_I,0)}} v\right).$$

(*c*) *Each vertex* $v \in T_I$ *induces a subtree*

$$T_v := (V_v, E_v), \quad V_v := \{w \in S^*(v)\}, \quad E_v := E \cap (V_v \times V_v),$$

which is an $\mathscr{H}$*-tree of the index set* v.

Proof. a) Use induction over the depth of $\mathscr{H}$-trees. b) Consider $T_I' := T_I \setminus \cup_{j=\ell+1}^{depth(T_I)}$ and apply part a). □

1.3.2 Block $\mathscr{H}$-Trees $T_{I\times J}$

For (rectangular) matrices from $\mathbb{R}^{I\times J}$ we need $\mathscr{H}$-trees with the root $I \times J$. The case $I \times I$ for square matrices is a particular subcase. Again, the concrete construction is postponed to §4.2.

Definition 1.11 (block $\mathscr{H}$-tree) *Let* I *and* J *be finite sets and let* T_I *and* T_J *be* $\mathscr{H}$*-trees of* I *and* J. *An* $\mathscr{H}$*-tree* $T_{I\times J}$ *is called a block* $\mathscr{H}$*-tree* (*based upon* T_I and T_J) *if for all* $v \in T_{I\times J}^{(\ell)}$ *there exist* $r \in T_I^{(\ell)}$ *and* $s \in T_J^{(\ell)}$ *such that* $v = r \times s$. *In the case* $T_I = T_J$ *we say that* $T_{I\times I}$ *is based on* T_I.

Given $v \in T_{I\times J}$, Definition 1.11 does not fix whether a vertex v is a leaf or not. But if $v = r \times s$ is not a leaf, the set of sons is given by $S(v) = \{v' = r' \times s' \mid r' \in S(r), s' \in S(s)\}$. Since by definition, r and s belong to some identical level number ℓ, the sons $v' = r' \times s' \in S(v)$ are products of r' and s' from level number $\ell + 1$. Furthermore, the set $T_{I\times J}^{(\ell)}$ defined in Definition 1.11 is a subset of $T_I^{(\ell)} \times T_J^{(\ell)}$.

Definition 1.12 (cardinality, submatrix, supermatrix) *Let* $M \in \mathbb{R}^{I\times J}$ *be a matrix over the index set* $I \times J$. *We denote the cardinality of a set* I *by* $\#I$. *The submatrix* $(M_{i,j})_{(i,j)\in I'\times J'}$ *for a subset* $I' \times J'$ *of* $I \times J$ *is denoted by* $M|_{I'\times J'}$. *For a superset* $I'' \times J'' \supset I \times J$ *we denote the matrix* $M'' \in \mathbb{R}^{I''\times J''}$ *with entries* $M''_{i,j} = \left\{\begin{array}{ll} M_{i,j} & \text{if } (i,j) \in I \times J \\ 0 & \text{otherwise} \end{array}\right\}$ by $M|^{I''\times J''}$.

Remark 1.13 (partitioning) *Due to Remark* 1.10*a*, *any block* $\mathscr{H}$*-tree* $T_{I\times J}$ *with root* $I \times J$ *has the property* $\dot\cup_{v\in\mathscr{L}(T_{I\times J})} = I \times J$. *Vice versa, given a partitioning* $\mathscr{P} \subset T_{I\times J}$

such that $\dot{\bigcup}_{v\in\mathscr{P}} = I \times J$, *there is a unique* $\mathscr{H}$*-subtree* $T'_{I\times J}$ *of* $T_{I\times J}$ (*with same root* $I \times J$) *so that* $\mathscr{L}(T'_{I\times J}) = \mathscr{P}$.

In previous papers (e.g., [8], [10]), we have based the $\mathscr{H}$-matrix on a partitioning $\mathscr{P} \subset T_{I\times J}$. Equivalently, we can use the associated $\mathscr{H}$-tree $T_{I\times J}$ with $\mathscr{L}(T'_{I\times J}) = \mathscr{P}$ (see Remark 1.13).

In principal, the $\mathscr{H}$-matrix uses the $R(k)$-representation (1.1) for all blocks $v = r \times s \in T_{I\times J}$. By practical reasons this is less efficient for small-sized blocks. Therefore, a minimal block size $n_{\min}$ will be introduced. The use of the $R(k)$-representation is restricted to $\min\{\#r, \#s\} > n_{\min}$, otherwise the standard full representation is used (in the later numerical examples we choose $n_{\min} = 32$).

1.3.3 Set of Hierarchical Matrices

Definition 1.14 ($\mathscr{H}$-matrix) *Let* $k, n_{\min} \in \mathbb{N}_0$. *The set of* $\mathscr{H}$*-matrices induced by a block* $\mathscr{H}$*-tree* T *with blockwise rank* k *and minimal block size* $n_{\min}$ *is defined as*

$$\mathscr{H}(T,k) := \{M \in \mathbb{R}^{I\times J} \mid \forall r \times s \in \mathscr{L}(T) : \operatorname{rank}(M|_{r\times s}) \leq k \text{ or } \#r \leq n_{\min} \text{ or } \#s \leq n_{\min}\}.$$

A matrix $M \in \mathscr{H}(T,k)$ *is said to be given in* $\mathscr{H}$*-matrix representation, if for all leaves* $r \times s$ *with* $\#r \leq n_{\min}$ *or* $\#s \leq n_{\min}$ *the corresponding matrix block* $M|_{r\times s}$ *is given in full matrix representation and in* $R(k)$*-matrix representation for the other leaves.*

2 $\mathscr{H}$-Matrix Arithmetics and Their Complexity

In the first part of this section we estimate the storage requirements of an $\mathscr{H}$-matrix, the cardinality of the $\mathscr{H}$-tree, the complexity of the matrix-vector multiplication, truncation and formatted addition of $\mathscr{H}$-matrices based on the *sparsity* of the $\mathscr{H}$-tree T. In the second part we define the idempotency constant which is needed to bound the complexity of the matrix multiplication and inversion in the set of $\mathscr{H}$-matrices.

2.1 Sparsity Based Estimates

Hierarchical matrices possess a certain kind of sparsity which is essential for favourable estimates of the storage and the cost of the matrix-vector multiplication and matrix addition.

2.1.1 Sparsity Constant and $\mathscr{H}$-Trees

The block $\mathscr{H}$-tree $T_{I\times J}$ may have a sparsity property which is measured by the quantity C_{sp} defined below. In §4.2, the construction of $T_{I\times J}$ will lead to block $\mathscr{H}$-trees with a sparsity constant C_{sp} independent of the size of $\#I$.

Definition 2.1 (sparsity constant) *Let $T_{I\times J}$ be a block $\mathscr{H}$-tree based on T_I and T_J. We define the sparsity (constant) C_{sp} of $T_{I\times J}$ by*

$$C_{\text{sp}} := \max\left\{\max_{r\in T_I} \#\{s \in T_J \mid r \times s \in T_{I\times J}\}, \max_{s\in T_J} \#\{r \in T_I \mid r \times s \in T_{I\times J}\}\right\}. \quad (2.1)$$

In many estimates (e.g., in (2.2) below) sums over the quantities $\#\{\ldots\}$ appear. Then the maximum C_{sp} from (2.1) could be replaced by the possibly smaller average.

In the following, we simplify the notation $T_{I\times J}$ by T without subscripts.

Lemma 2.2 (*a*) *Let T be a block $\mathscr{H}$-tree based on T_I and T_J with sparsity constant C_{sp}. If T_I and T_J satisfy $\#S(v) \neq 1$ for all vertices $v \in T_I \cup T_J$, then*

$$\#T_I \leq 2\#I, \quad \#T \leq 2C_{\text{sp}} \min\{\#I, \#J\}.$$

(*b*) *Let* $p := \text{depth}(T) \geq 1$. If $\#S(v) \neq 1$ *is not necessarily fulfilled, then it still holds*

$$\#T_I \leq 2p\#I, \quad \#T \leq 2pC_{\text{sp}} \min\{\#I, \#J\}.$$

(*c*) *The previous estimates provide a bound for $\#\mathscr{L}(T) \leq \#T$.*

Proof. The first inequality of part (a) is trivial. The second inequality is derived by

$$\#T = \sum_{r\times s\in T} 1 = \sum_{r\in T_I} \#\{r \times s \in T\} \leq \sum_{r\in T_I} C_{\text{sp}} \leq 2\#IC_{\text{sp}}. \quad (2.2)$$

Part (c) is a consequence of $\mathscr{L}(T) \subset T$. □

Due to the distinction between the $R(k)$-representation and the full representation, we introduce $\mathscr{L}^-(T)$ and $\mathscr{L}^+(T)$.

Definition 2.3 *Let T be a block $\mathscr{H}$-tree based on T_I and T_J. The set of "small" leaves of T is denoted by $\mathscr{L}^-(T) := \{r \times s \in \mathscr{L}(T) \mid \#r \leq n_{\min}$ or $\#s \leq n_{\min}\}$ and the set of "large" leaves is denoted as $\mathscr{L}^+(T) := \mathscr{L}(T) \setminus \mathscr{L}^-(T)$.*

Later, in (3.6), it will turn out that $n_{\min}$ should not be smaller than a constant given there.

2.1.2 Storage

The estimate in the next lemma makes use of the set of occupied levels L of a block $\mathscr{H}$-tree T defined by

$$L := \{i \in \mathbb{N}_0 \mid \mathscr{L}(T, i) \neq \emptyset\}. \quad (2.3)$$

In particular, $\#L$ is of interest. Note that $\#L \leq \text{depth}(T) + 1$.

Lemma 2.4 (storage) *Let T be a block $\mathscr{H}$-tree based on T_I and T_J with sparsity constant C_{sp} (cf. (2.1)) and minimal block size $n_{\min}$. Then the storage requirements $N_{\mathscr{H},St}(T,k)$ for an $\mathscr{H}$-matrix $M \in \mathscr{H}(T,k)$ are bounded by*

$$N_{\mathscr{H},St}(T,k) \leq \#LC_{\text{sp}} \max\{k, n_{\min}\}(\#I + \#J).$$

Proof.

$$\begin{aligned}
N_{\mathscr{H},St}(T,k) &\overset{\text{Def.1.14}}{=} \sum_{r\times s\in\mathscr{L}^-(T)} N_{F,St}(\#r,\#s) \quad + \sum_{r\times s\in\mathscr{L}^+(T)} N_{R,St}(\#r,\#s,k) \\
&\overset{\text{Rem.1.2}}{\leq} \sum_{r\times s\in\mathscr{L}^-(T)} n_{\min}(\#r+\#s) \quad + \sum_{r\times s\in\mathscr{L}^+(T)} k(\#r+\#s) \qquad (2.4) \\
&\leq \sum_{r\times s\in\mathscr{L}(T)} \max\{k,n_{\min}\}\#r \quad + \sum_{r\times s\in\mathscr{L}(T)} \max\{k,n_{\min}\}\#s \\
&\overset{\text{Def.2.1}}{\leq} \sum_{i\in L}\sum_{r\in T_I^{(i)}} C_{\text{sp}} \max\{k,n_{\min}\}\#r \quad + \quad \sum_{i\in L}\sum_{s\in T_J^{(i)}} C_{\text{sp}} \max\{k,n_{\min}\}\#s \\
&\overset{\text{Rem.1.10}}{\leq} \sum_{i\in L} C_{\text{sp}} \max\{k,n_{\min}\}\#I \quad + \quad \sum_{i\in L} C_{\text{sp}} \max\{k,n_{\min}\}\#J \\
&= \quad \#LC_{\text{sp}} \max\{k,n_{\min}\}(\#I+\#J).
\end{aligned}$$

□

In line (2.4), the maximum in $N_{F,St}(\#r,\#s) \leq n_{\min} * \max\{\#r,\#s\}$ is estimated by $\#r + \#s$. Under the assumption $\#r \approx \#s$, this is an overestimation by the factor 2. Therefore, $\#LC_{\text{sp}} \max\{k, \frac{1}{2}n_{\min}\}(\#I + \#J)$ is supposed to be closer to $N_{\mathscr{H},St}(T,k)$.

The aim will be to construct T such that $\text{depth}(T) = \mathscr{O}(\log n)$, where n is the size of I and J.

2.1.3 Matrix-Vector Multiplication

Lemma 2.5 (matrix-vector product) *Let T be a block $\mathscr{H}$-tree. The complexity $N_{\mathscr{H}\cdot v}(T,k)$ of the matrix-vector product in the set of $\mathscr{H}$-matrices can be bounded from above and below by*

$$N_{\mathscr{H},St}(T,k) \leq N_{\mathscr{H}\cdot v}(T,k) \leq 2N_{\mathscr{H},St}(T,k).$$

Proof. According to Remark 1.2 the storage requirements in a block $r \times s$ in full matrix representation are $\#r\#s$. The cost to multiply the submatrix with a vector x and add the result to the target vector y are $2\#r\#s - \#r$ for the multiplication and $\#r$ for the addition:

$$N_{F,St} \leq N_{F\cdot v} \leq 2N_{F,St}.$$

For a block $r \times s$ in $R(k)$-matrix representation the storage requirements are $k(\#r + \#s)$. The cost to multiply the submatrix with a vector x and add the result to the target vector y are (due to Remark 1.2) $2k(\#r + \#s) - \#r - k$ for the multiplication and $\#r$ for the addition:

$$N_{R,St} \leq N_{R\cdot v} \leq 2N_{R,St}.$$

Since an $\mathscr{H}$-matrix consists blockwise of either full matrices or $R(k)$-matrices, this concludes the proof. $\square$

Algorithm 2.6 (matrix-vector product) *Let $M \in \mathscr{H}(T,k)$ be an $\mathscr{H}$-matrix. To compute the matrix-vector product $y := y + Mx$ with $x \in \mathbb{R}^J, y \in \mathbb{R}^I$, we call* MVM $(M, I \times J, x, y)$, *where* MVM *is the following procedure*:

```
procedure MVM(M, r × s, x, var y);
begin
    if S(r × s) ≠ ∅ then                                  {subdivide block}
        for each r' × s' ∈ S(r × s) do MVM(M, r' × s', x, y)
    else y|_r := y|_r + M|_{r×s} x|_s                     {full or R(k) − matrix}
end;
```

2.1.4 Truncation

In (1.3), we have defined the truncation $\mathscr{T}^R_{k' \leftarrow k}$ of $R(k)$-matrices. The extension to $\mathscr{H}$-matrices is given below.

Definition 2.7 (truncation of $\mathscr{H}$-matrices) *Let T be a block $\mathscr{H}$-tree and let $k, k' \in \mathbb{N}_0$. We define the truncation operator*

$$\mathscr{T}^{\mathscr{H}}_{k' \leftarrow k} : \mathscr{H}(T,k) \to \mathscr{H}(T,k')$$

by $M' = \mathscr{T}^{\mathscr{H}}_{k' \leftarrow k}(M)$ *with* $M'|_{r\times s} = \mathscr{T}^R_{k' \leftarrow k}(M|_{r\times s})$ *for all* $r \times s \in \mathscr{L}^+(T)$ *and* $M'|_{r\times s} = M|_{r\times s}$ *for all* $r \times s \in \mathscr{L}^-(T)$.

Remark 2.8 $\mathscr{T}^{\mathscr{H}}_{k' \leftarrow k}$ *maps a matrix $M \in \mathscr{H}(T,k)$ to a best approximation $M' \in \mathscr{H}(T,k')$ of M with respect to the Frobenius norm. Since there is possibly more than one best approximation we choose an arbitrary representative.*

Note that $\#\mathscr{L}(T)$ appearing in the next estimate can be bounded by means of Lemma 2.2c.

Lemma 2.9 (complexity of the $\mathscr{H}$-matrix truncation) *Let T be a block $\mathscr{H}$-tree based on the $\mathscr{H}$-trees T_I and T_J. A truncation $\mathscr{T}^{\mathscr{H}}_{k' \leftarrow k}(M)$ of an $\mathscr{H}$-matrix $M \in \mathscr{H}(T,k)$ can be computed with complexity*

$$N_{\mathscr{H},k' \leftarrow k}(T) \leq 6kN_{\mathscr{H},St}(T,k) + 23k^3 \#\mathscr{L}(T).$$

Proof. Lemma 1.3b and Remark 1.2 show

$$N_{\mathscr{H},k'\leftarrow k}(T) = \sum_{r\times s\in\mathscr{L}^+(T)} N_{R,\mathrm{SVD}}(\#r,\#s,k) \overset{\text{Lemma1.3b}}{\leq} \sum_{r\times s\in\mathscr{L}^+(T)} 6k^2(\#r+\#s)+23k^3$$

$$= 6k\Big[\sum_{r\times s\in\mathscr{L}^+(T)} k(\#r+\#s)\Big] + 23k^3\#\mathscr{L}^+(T) \overset{(1.2)}{\leq} 6kN_{\mathscr{H},St}(k,T)+23k^3\#\mathscr{L}(T).$$

□

A sum M of q $R(k)$-matrices A_i $(1\leq i\leq q)$ is an $R(qk)$-matrix. Instead of the optimal truncation $M' = \mathscr{T}^{\mathscr{R}}_{k\leftarrow qk}(\sum A_i)$, we can apply the cheaper $\mathscr{T}^{\mathscr{R}}_{k\leftarrow 2k}$-truncation to sums of only two terms: $M_2 := \mathscr{T}^{\mathscr{R}}_{k\leftarrow 2k}(A_1+A_2)$, $M_i := \mathscr{T}^{\mathscr{R}}_{k\leftarrow 2k}(M_{i-1}+A_i)$ $(i=3,\dots,q)$ resulting in $M'' := M_q$ (in general, $M''\neq M'$).

Lemma 2.10 (fast truncation of $\mathscr{H}$-matrices) *Let T be a block $\mathscr{H}$-tree. An approximate truncation of an $\mathscr{H}$-matrix from $\mathscr{H}(T,qk)$ to $\mathscr{H}(T,k)$ (not necessarily a best approximation) can be computed with complexity*

$$N_{\mathscr{H},k\overset{\text{fast}}{\leftarrow}qk}(T) \leq (q-1)(24kN_{\mathscr{H},St}(T,k)+184k^3\#\mathscr{L}(T))$$

by successive use of the truncation $\mathscr{T}^{\mathscr{H}}_{k\leftarrow 2k}$: let $M\in\mathscr{H}(T,qk)$ be decomposed into $M=\sum_{i=1}^{q}M_i$ with matrices $M_i\in\mathscr{H}(T,k)$. Then we define

$$\tilde{M}_1 := M_1 \quad\text{and}\quad \tilde{M}_j := \mathscr{T}^{\mathscr{H}}_{k\leftarrow 2k}(\tilde{M}_{j-1}+M_j) \quad\text{for } j=2,\dots,q.$$

The matrix $\tilde{M}_q$ is the desired approximation in $\mathscr{H}(T,k)$.

The truncation procedure from Lemma 2.9 is useful for theoretical purposes because it computes a best approximation. The fast truncation procedure from Lemma 2.10 can yield arbitrarily poor results (because of cancellation of the singular values), but in practice this is not likely to occur.

If we want to approximate an $\mathscr{H}(T,k)$-matrix by an $R(k)$-matrix then we can exploit the hierarchical structure of the $\mathscr{H}$-matrix format to do this with almost linear complexity. This is by itself an important result, but we will also use this (fast) conversion in the multiplication procedure for $\mathscr{H}$-matrices in Section 2.2.2.

Algorithm 2.11 (hierarchical conversion) *Let T be a block $\mathscr{H}$-tree of depth $p:=\mathrm{depth}(T)$ where each vertex $v\in T$ has at most C_{sons} successors. For a matrix $M\in\mathscr{H}(T,k)$ we compute an approximation $R_{\mathscr{H}}\in R(k)$ in $p+1$ steps:*

1. *We convert the matrix blocks of M corresponding to "small" leaves $r\times s\in\mathscr{L}^-(T)$ to $R(k)$-format and retain the "large" leaves $r\times s\in\mathscr{L}^-(T)$:*

$$R_p|_{r\times s} := \begin{cases} \mathscr{T}^{\mathscr{R}}_{k\leftarrow n_{\min}}(M|_{r\times s}) & \text{if } r\times s\in\mathscr{L}^-(T),\\ M|_{r\times s} & \text{otherwise.}\end{cases}$$

2. *For each* $\ell = p-1,\ldots,0$ *we define the matrix* R_ℓ *blockwise for all* $r\times s \in T^{(\ell)} \cup \mathscr{L}(T,\ell-1)\cup\cdots\cup\mathscr{L}(T,0)$ (*cf. Remark* 1.10) *by*

$$R_\ell|_{r\times s} := \begin{cases} \mathscr{T}^{\mathscr{R}}_{k\leftarrow C_{\text{sons}}k}(R_{\ell+1}|_{r\times s}) & \text{if } r\times s\in T^{(\ell)},\\ R_{\ell+1}|_{r\times s} & \text{otherwise.}\end{cases}$$

The last matrix $R_0 \in R(k)$ *is the desired approximation* $R_{\mathscr{H}}$ *to* M.

Lemma 2.12 (accuracy and complexity of the hierarchical conversion) *We use the notation from Algorithm* 2.11. *If* R_{best} *denotes an* $R(k)$*-best approximation to* M (*with respect to the Frobenius norm*) *and* $R_{\mathscr{H}}$ *the above defined hierarchical approximation, then the error is bounded by*

$$\|R_{\mathscr{H}} - M\|_F \le (2^{p+1}+1)\|R_{\text{best}} - M\|_F$$

while the complexity for the conversion (*we assume* $n_{\min}\le k$ *and* $C_{\text{sons}}\ge 2$) *is*

$$N_{R(k)\leftarrow\mathscr{H}} \le 6C_{\text{sp}}C_{\text{sons}}^2k^2(p+1)(\#I+\#J) + 23C_{\text{sons}}^3k^3\#T.$$

Proof. a) (Complexity) The conversion of the full matrix blocks $r\times s\in\mathscr{L}^-(T)$ to $R(k)$-format is done by a singular value decomposition which has a complexity of $21n_{\min}^3$. For all vertices $r\times s\in T\setminus\mathscr{L}(T)$ we have to truncate the sum over all sons of $r\times s$, which due to Remark 1.3a is of complexity $6C_{\text{sons}}^2k^2(\#r+\#s)+23C_{\text{sons}}^3k^3$. For all vertices this sums up to

$$\begin{aligned} N_{R(k)\leftarrow\mathscr{H}} &\le \sum_{r\times s\in\mathscr{L}^-(T)} 21n_{\min}^3 + \sum_{r\times s\in T\setminus\mathscr{L}(T)}\left(6C_{\text{sons}}^2k^2(\#r+\#s)+23C_{\text{sons}}^3k^3\right)\\ &\le \sum_{r\times s\in T}\left(6C_{\text{sons}}^2k^2(\#r+\#s)+23C_{\text{sons}}^3k^3\right)\\ &\le (p+1)C_{\text{sp}}6C_{\text{sons}}^2k^2(\#I+\#J)+23C_{\text{sons}}^3k^3\#T.\end{aligned}$$

b) (Error) We define the sets

$$\mathscr{R}(T,\ell,k) := \{M\in\mathbb{R}^{I\times J}\mid \forall r\times s\in T^{(\ell)}\cup\mathscr{L}(T,\ell-1)\cup\cdots\cup\mathscr{L}(T,0) : \operatorname{rank}(M|_{r\times s})\le k\}.$$

By R_ℓ we denote the matrix appearing in the ℓth step of the algorithm. Obviously R_ℓ is contained in the set $\mathscr{R}(T,\ell,k)$. The matrix R_0 is the resulting approximant $R_{\mathscr{H}}$. From one level ℓ to the next level $\ell-1$, the algorithm determines a best approximation (with respect to the Frobenius norm) of the matrix R_ℓ in the set $\mathscr{R}(T,\ell,k)$:

$$\forall \tilde{R}\in\mathscr{R}(T,\ell,k): \quad \|R_\ell - R_{\ell-1}\|_F \le \|R_\ell-\tilde{R}\|_F. \tag{2.5a}$$

In the first step (conversion of the full matrix blocks) this reads

$$\forall \tilde{R}\in\mathscr{R}(T,p,k): \quad \|M - R_p\|_F \le \|M-\tilde{R}\|_F. \tag{2.5b}$$

By induction we prove $\|R_\ell - R_{\text{best}}\|_F \le 2^{p-\ell}\|R_p - R_{\text{best}}\|_F$ as follows. The start $\ell = p$ of the induction is trivial. The induction step $\ell \mapsto \ell - 1$ follows from

$$\|R_{\ell-1} - R_{\text{best}}\|_F \le \|R_{\ell-1} - R_\ell\|_F + \|R_\ell - R_{\text{best}}\|_F \overset{(2.5a)}{\le} 2\|R_{\ell-1} - R_{\text{best}}\|_F.$$

Using this inequality, we can conclude that

$$\begin{aligned}
\|M - R_0\|_F &= \|M - \sum_{\ell=0}^{p-1}(R_\ell - R_{\ell+1}) - R_p\|_F \le \|M - R_p\|_F + \sum_{\ell=0}^{p-1}\|R_\ell - R_{\ell+1}\|_F \\
&\overset{(2.5a,b)}{\le} \|M - R_{\text{best}}\|_F + \sum_{\ell=0}^{p-1}\|R_{\text{best}} - R_{\ell+1}\|_F \\
&\le \|M - R_{\text{best}}\|_F + \sum_{\ell=0}^{p-1} 2^{p-\ell-1}\|R_p - R_{\text{best}}\|_F \le 2^p\|R_p - R_{\text{best}}\|_F + \|M - R_{\text{best}}\|_F \\
&\le 2^p(\|R_p - M\|_F + \|M - R_{\text{best}}\|_F) + \|M - R_{\text{best}}\|_F \overset{(2.5b)}{\le} (2^{p+1}+1)\|M - R_{\text{best}}\|_F.
\end{aligned}$$

□

2.1.5 Addition

Definition 2.13 (formatted $\mathcal{H}$-matrix addition) *The formatted addition $\oplus : \mathcal{H}(T,k) \times \mathcal{H}(T,k) \to \mathcal{H}(T,k)$ is defined as a truncation of the (exact) sum to the set of $\mathcal{H}$-matrices, i.e., $A \oplus B := \mathcal{T}^{\mathcal{H}}_{k \leftarrow 2k}(A+B)$.*

Remark 2.14 *According to Lemma* 2.9 *the complexity of the formatted $\mathcal{H}$-matrix addition is bounded by*

$$N_{\mathcal{H},\oplus}(T,k) \le 24kN_{\mathcal{H},St}(T,k) + 184k^3\#\mathcal{L}(T).$$

In the later inversion procedure (see Table 1) we have to add three $\mathcal{H}$-matrices A, B, C and to truncate the sum to rank k and to overwrite C by the result. This is done by the following algorithm.

Algorithm 2.15 (formatted $\mathcal{H}$-matrix addition) *Let $A, B, C \in \mathcal{H}(T,k)$ be $\mathcal{H}$-matrices over the index set $I \times J$. To compute the (formatted) sum $C := \mathcal{T}^{\mathcal{H}}_{k \leftarrow 3k}(A+B+C)$ we use the following procedure (called by Add(C, $I \times J$, A, B)):*

```
procedure Add(var C, r × s, A, B);
begin
    if S(r × s) ≠ ∅ then                                  {subdivide block}
      for each r' × s' ∈ S(r × s) do Add(C, r' × s', A, B)
    else C|_{r×s} := T^R_{k←3k}(C|_{r×s} + A|_{r×s} + B|_{r×s})     {full or R(k) − matrix}
end;
```

2.1.6 Matrix-Matrix Multiplication

We consider the multiplication of two (rectangular) matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{J \times K}$ (a particular case is $I = J = K$). To elucidate the difficulty of the multiplication, we recall that the addition is a structure-preserving operation in the sense that the sum of two $\mathscr{H}$-matrices based on the $\mathscr{H}$-tree T can be represented using the same $\mathscr{H}$-tree T and the sum of the blockwise ranks. In contrast to the addition, the product of two $\mathscr{H}$-matrices is much more complicated: even if $I = J = K$ and if A and B belong to the same set $\mathscr{H}(T, k)$, the tree T is in general not suitable for the representation of the (exact) product. A suitable tree is the product tree $T \cdot T$, which is defined next.

Definition 2.16 (product of block $\mathscr{H}$-trees) Let $T = T_{I \times J}$ be a block $\mathscr{H}$-tree based on T_I, T_J and let $T' = T_{J \times K}$ be a block $\mathscr{H}$-tree based on T_J, T_K. We define the product tree $T_{I \times K}$ (denoted by $T \cdot T'$) by means of $\operatorname{root}(T \cdot T') := I \times K$ and the description of the set of sons of each node. For each level $\ell = 0, \ldots, p-1$ and each vertex $r \times t \in (T \cdot T')^{(\ell)}$, the set of sons of $r \times t$ is defined by

$$S(r \times t) := \left\{ r' \times t' \,\middle|\, \exists s \in T_J^{(\ell)} \exists s' \in T_J^{(\ell+1)} : r' \times s' \in S_T(r \times s),\ s' \times t' \in S_{T'}(s \times t) \right\}.$$

We remark that depth $(T \cdot T') \leq \min\{\operatorname{depth}(T), \operatorname{depth}(T')\}$.

Lemma 2.17 (*a*) *Let T be a block $\mathscr{H}$-tree based on T_I, T_J and let T' be a block $\mathscr{H}$-tree based on T_J, T_K. Then the tree $T \cdot T'$ is a block $\mathscr{H}$-tree based on T_I, T_K.*

(*b*) *Let $C_{\text{sp}}(T)$ and $C_{\text{sp}}(T')$ denote the corresponding sparsity constant. Then the sparsity of $T \cdot T'$ can be estimated by*

$$C_{\text{sp}}(T \cdot T') \leq C_{\text{sp}}(T) C_{\text{sp}}(T').$$

Proof. Let $r \in T_I$. Due to the symmetry of the sparsity we only give a bound for $\#\{t \in T_K \mid r \times t \in T \cdot T'\}$:

$$\{t \in T_K \mid r \times t \in T \cdot T'\} \overset{\text{Def.2.16}}{\subset} \{t \in T_K \mid \exists s \in T_J : r \times s \in T, s \times t \in T'\},$$

$$\#\{t \in T_K \mid r \times t \in T \cdot T'\} \leq \sum_{s \in T_J, r \times s \in T} \#\{t \in T_K \mid s \times t \in T'\} \quad \leq \quad C_{\text{sp}}(T) C_{\text{sp}}(T').$$

□

Definition 2.18 (predecessors) *Let T be an $\mathscr{H}$-tree, $i \in [0, \operatorname{depth}(T)]$, $t \in T^{(i)}$. We define the predecessor of t on level $j \in \{0, \ldots, i\}$ as the uniquely determined vertex $\mathscr{F}^j(t) \in T^{(j)}$ with $t \in S^*(\mathscr{F}^j(t))$.*

Due to the $\mathscr{H}$-tree property, the condition $t \in S^*(\mathscr{F}^j(t))$ can equivalently be defined by $t \subset \mathscr{F}^j(t)$.

In the following, we describe the *exact* multiplication of two $\mathscr{H}$-matrices. In § 2.2 we consider the truncation to a given format and rank, which leads to the formatted multiplication $\odot$ analogously to the formatted addition $\oplus$ from § 2.1.5.

Lemma 2.19 (representation of the $\mathscr{H}$-matrix product) *Let T be a block $\mathscr{H}$-tree based on T_I, T_J and let T' be a block $\mathscr{H}$-tree based on T_J, T_K. For each leaf $r \times t \in \mathscr{L}(T \cdot T', i)$ we define*

$$\mathscr{U}(r \times t, j) := \left\{ s \in T_J^{(j)} \middle| \begin{array}{l} \mathscr{F}^j(r) \times s \in T \text{ and } s \times \mathscr{F}^j(t) \in T' \text{ and} \\ (\mathscr{F}^j(r) \times s \in \mathscr{L}(T) \text{ or } s \times \mathscr{F}^j(t) \in \mathscr{L}(T')) \end{array} \right\}, \quad j \in \mathbb{N}_0.$$

Then for two matrices $M \in \mathscr{H}(T, k)$ and $M' \in \mathscr{H}(T', k')$ and each $r \times t \in \mathscr{L}(T \cdot T', i)$ there holds

$$(M \cdot M')|_{r \times t} = \sum_{j=0}^{i} \sum_{s \in \mathscr{U}(r \times t, j)} M|_{r \times s} M'|_{s \times t} \tag{2.6}$$

and

$$J = \dot{\bigcup_{j=0,\ldots,i}} \dot{\bigcup_{s \in \mathscr{U}(r \times t, j)}} s. \tag{2.7}$$

Proof. a) Assuming that (2.7) is true, we conclude the representation formula (2.6) from (2.7). The proof of (2.7) is given in the following parts b–d.

b) (Disjointness of $\mathscr{U}(r \times t, j)$) According to Remark 1.10b, the elements of $\mathscr{U}(r \times t, j)$ are disjoint.

c) (Disjointness w.r.t. j) Let $s \in \mathscr{U}(r \times t, j)$, $s' \in \mathscr{U}(r \times t, j')$, $j \leq j'$ and $s \cap s' \neq \emptyset$. Since $s, s' \in T_J$ and T_J is an $\mathscr{H}$-tree we get $s' \subset s$, $\mathscr{F}^j(s') = s$. It follows

$$\mathscr{F}^{j'}(r) \times s' \subset \mathscr{F}^j(r) \times s, \qquad s' \times \mathscr{F}^{j'}(t) \subset s \times \mathscr{F}^j(t). \tag{2.8}$$

Due to the definition of $\mathscr{U}(r \times t, i)$ either $\mathscr{F}^j(r) \times s$ or $s \times \mathscr{F}^j(t)$ is a leaf. Hence, one inclusion in (2.8) becomes an equality which implies $j' = j$.

d) (Covering) Let $j \in J$. It holds $t_0 := \mathscr{F}^{(0)}(r) \times J \in T$, $t'_0 := J \times \mathscr{F}^{(0)}(t) \in T'$ and $j \in J$. If neither t_0 nor t'_0 is a leaf, then there exists $J' \in S(J)$ such that $j \in J'$ and $t_1 := \mathscr{F}^{(1)}(r) \times J' \in T$, $t'_1 := J' \times \mathscr{F}^{(1)}(t) \in T'$. By induction we define $t_i = \mathscr{F}^{(i)}(r) \times s$, $t'_i = s \times \mathscr{F}^{(i)}(t)$ with $j \in s$. Let i be the first index for which either $t_i = \mathscr{F}^{(i)}(r) \times s$ or $t'_i = s \times \mathscr{F}^{(i)}(t)$ is a leaf. Then $j \in s \in \mathscr{U}(r \times t, i)$. □

Theorem 2.20 (structure of the $\mathscr{H}$-matrix product) *Let T be a block $\mathscr{H}$-tree based on T_I, T_J and let T' be a block $\mathscr{H}$-tree based on T_J, T_K. Let $C_{\mathrm{sp}}(T)$ and $C_{\mathrm{sp}}(T')$ denote the sparsity constant of T and T' and set $p := \min\{\mathrm{depth}(T), \mathrm{depth}(T')\}$. The exact* *multiplication is a mapping* $\cdot : \mathscr{H}(T,k) \times \mathscr{H}(T',k') \to \mathscr{H}(T \cdot T', \tilde{k})$ *for some* $\tilde{k}$ *which can be bounded by*

$$\tilde{k} \le (p+1)\min\{C_{\mathrm{sp}}(T), C_{\mathrm{sp}}(T')\}\max\{k,k',n_{\min}\}. \tag{2.9}$$

The exact multiplication can be performed with complexity

$$N_{\mathscr{H},\cdot}(T,T') \le 2(p+1)C_{\mathrm{sp}}(T)C_{\mathrm{sp}}(T')\big(\max\{k',n_{\min}\}N_{\mathscr{H},St}(T,k) + \max\{k,n_{\min}\}N_{\mathscr{H},St}(T',k')\big).$$

Proof. a) (Rank) Let $M \in \mathscr{H}(T,k)$, $M' \in \mathscr{H}(T',k')$, and $r \times t \in \mathscr{L}(T \cdot T')$. Due to (2.6), we can express the product by $(p+1)\max_{j=0}^{i} \#\mathscr{U}(r \times t, j)$ addends, each of which is a product of two matrices. From the definition of $\mathscr{U}(r \times t, j)$ we get that for each addend one of the factors corresponds to a leaf and so its rank is bounded by $\max\{k,k',n_{\min}\}$. Hence, each addend has a rank bounded by $\max\{k,k',n_{\min}\}$. It follows that $\tilde{k} \le (p+1)\max_{j=0}^{i} \#\mathscr{U}(r \times t, j)\max\{k,k',n_{\min}\}$. The cardinality of $\mathscr{U}(r \times t, j)$ is bounded by

$$\#\mathscr{U}(r \times t, j) \le \#\{s \in T_J^{(j)} \mid \mathscr{F}^j(r) \times s \in T\} \le C_{\mathrm{sp}}(T),$$
$$\#\mathscr{U}(r \times t, j) \le \#\{s \in T_J^{(j)} \mid s \times \mathscr{F}^j(t) \in T'\} \le C_{\mathrm{sp}}(T'),$$

which yields $\#\mathscr{U}(r \times s, j) \le \min\{C_{\mathrm{sp}}(T), C_{\mathrm{sp}}(T')\}$.

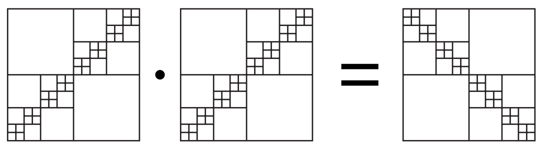
b) (Complexity) Using the representation formula (2.6), we have to compute the products $M|_{r\times s}M'|_{s\times t}$ that consist (due to the definition of $\mathscr{U}(r \times t, j)$) of $\max\{k,k',n_{\min}\}$ matrix-vector products. In the following, the expressions $N_{\mathscr{H},St}(T_{r\times J},k)$ and $N_{\mathscr{H},St}(T'_{J\times t},k')$ appear which denote the storage requirements for a submatrix to the index set $r \times J$ and $J \times t$ of a matrix in $\mathscr{H}(T,k)$ and $\mathscr{H}(T',k')$. We use the abbreviations $\kappa := \max\{k,n_{\min}\}$ and $\kappa' := \max\{k',n_{\min}\}$ and conclude that

$$\begin{aligned}
N_{\mathscr{H},\cdot}(T,T') &\overset{\text{Lem.2.5}}{\le} \sum_{r\times t \in \mathscr{L}(T\cdot T')} \sum_{j=0}^{p} \sum_{s \in \mathscr{U}(r\times t,j)} \max\{2\kappa' N_{\mathscr{H},St}(T_{r\times s},k), 2\kappa N_{\mathscr{H},St}(T'_{s\times t},k')\} \\
&\overset{(2.7)}{\le} \sum_{r\times t \in \mathscr{L}(T\cdot T')} 2\max\{\kappa' N_{\mathscr{H},St}(T_{r\times J},k), \kappa N_{\mathscr{H},St}(T'_{J\times t},k')\} \\
&= 2\sum_{i=0}^{p}\left(\sum_{r\times t \in \mathscr{L}(T\cdot T',i)} \kappa' N_{\mathscr{H},St}(T_{r\times J},k) \quad + \sum_{r\times t \in \mathscr{L}(T\cdot T',i)} \kappa N_{\mathscr{H},St}(T'_{J\times t},k')\right) \\
&\overset{\text{Lem.2.17}}{\le} 2\sum_{i=0}^{p}\big(C_{\mathrm{sp}}(T)C_{\mathrm{sp}}(T')\kappa' N_{\mathscr{H},St}(T,k) + C_{\mathrm{sp}}(T)C_{\mathrm{sp}}(T')\kappa N_{\mathscr{H},St}(T',k')\big) \\
&\le 2(p+1)C_{\mathrm{sp}}(T)C_{\mathrm{sp}}(T')\big(\kappa' N_{\mathscr{H},St}(T,k) + \kappa N_{\mathscr{H},St}(T',k')\big),
\end{aligned}$$

proving the last estimate. □

The factor $p+1$ in (2.9) can be replaced by $\#L$ with L corresponding to $\mathscr{L}(T \cdot T')$ (cf. (2.3)).

Remark 2.21 *Lemma* 2.17 *shows that the product of two sparse $\mathscr{H}$-matrices will always yield a sparse $\mathscr{H}$-matrix. Theorem* 2.20 *bounds the blockwise rank of the product. However, the product tree $T \cdot T'$ may change drastically even if $T = T'$:*



2.2 Idempotency Based Estimates

2.2.1 Case $I = J = K$

Next, we consider the case $I = J = K$. Given matrices $A, B \in \mathscr{H}(T, k)$, where T is the block $\mathscr{H}$-tree based on T_I, we would like to get a product AB in $\mathscr{H}(T, k)$. Due to § 2.1.6, the result is a matrix in $\mathscr{H}(T \cdot T, \tilde{k})$ with the product tree $T \cdot T$ instead of T. The necessary conversion from $\mathscr{H}(T \cdot T, \tilde{k})$ into $\mathscr{H}(T, k'')$ is discussed in the following.

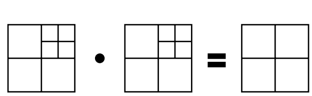
An $\mathscr{H}$-tree T may be called idempotent if $T \cdot T = T$ holds for the multiplication of Definition 2.16. In that case, we immediately get the desired representation formula (2.6) of the product of two $\mathscr{H}$-matrices from $\mathscr{H}(T, k)$ in the same set. In general, however, the tree T will not be idempotent but *almost* idempotent, which will be measured by the *idempotency constant* introduced below.

Definition 2.22 (idempotency) *Let T be a block $\mathscr{H}$-tree based on T_I. We define the elementwise idempotency $C_{\mathrm{id}}(r \times t)$ and idempotency constant $C_{\mathrm{id}}(T)$ by*

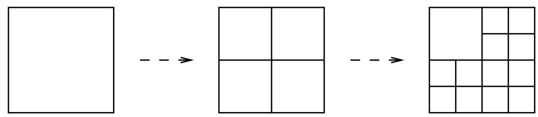
$$
\begin{aligned}
C_{\mathrm{id}}(r \times t) &:= \#\{r' \times t' \mid r' \in S^*(r), t' \in S^*(t) \text{ and } \exists s' \in T_I : r' \times s' \in T, s' \times t' \in T\},\\
C_{\mathrm{id}}(T) &:= \max_{r \times t \in \mathscr{L}(T)} C_{\mathrm{id}}(r \times t).
\end{aligned}
$$

If the tree T is fixed, the short notation C_{id} is used instead of $C_{\mathrm{id}}(T)$.

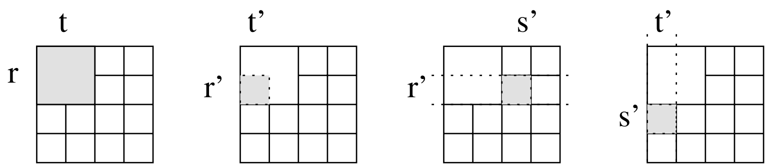
If the tree T is idempotent, then for any $r \times t \in \mathscr{L}(T)$ and $s \in T_I$ there holds $r \times s \in \mathscr{L}(T)$ or $s \times t \in \mathscr{L}(T)$ (see Definition 2.16) so that $C_{\mathrm{id}} = 1$. The reverse statement is not true: if $C_{\mathrm{id}} = 1$ then T is not necessarily idempotent, because the tree $T \cdot T$ can be coarser than T:



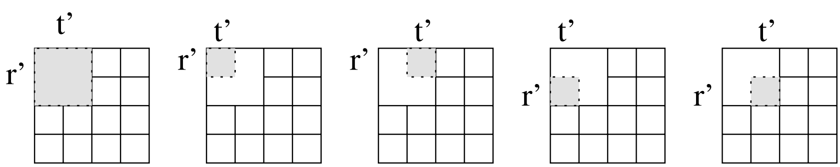
Example 2.23 *To illustrate Definition 2.22 we consider the block* $\mathscr{H}$*-tree*



and the leaf $r \times t$ *in the top left corner*:



The vertex s' *connects the two vertices* r', t' *in the sense that* $r' \times s' \in T$ *and* $s' \times t' \in T$, *while* $r' \times t' \notin T$. *The number of vertices* r', t' *that are contained in* r, t *and connected by a vertex* s' *is* 5, *which is the elementwise idempotency* $C_{\mathrm{id}}(r \times t)$ *of the vertex* $r \times t$:



The following theorem provides a matrix product such that the result lies in $\mathscr{H}(T, k'')$ (same tree T as for the factors).

Theorem 2.24 (multiplication of $\mathscr{H}$-matrices) *Let* T *be a block* $\mathscr{H}$*-tree of the index set* $I \times I$ *with idempotency constant* C_{id}, *sparsity constant* C_{sp} *and depth* p. *We assume* (*for simplicity*) $n_{\min} \leq k, k'$. *The exact multiplication is a mapping* $\cdot : \mathscr{H}(T, k) \times \mathscr{H}(T, k') \to \mathscr{H}(T, \tilde{k})$ *with some* $\tilde{k}$ *bounded by*

$$\tilde{k} \leq C_{\mathrm{id}} C_{\mathrm{sp}} (p + 1) \max\{k, k'\}.$$

The formatted multiplication $\odot^{\mathrm{best}} : \mathscr{H}(T, k) \times \mathscr{H}(T, k') \to \mathscr{H}(T, k'')$ *for any* $k'' < \tilde{k}$ *is defined as the exact multiplication followed by the truncation* $\mathscr{T}^{\mathscr{H}}_{k'' \leftarrow \tilde{k}}$ *of Lemma* 2.9 *and can be computed with complexity*

$$N_{\mathscr{H}, \odot, \mathrm{best}}(T, k, k') \leq 43 C_{\mathrm{id}}^3 C_{\mathrm{sp}}^3 k^3 (p + 1)^3 \max\{\#I, \#\mathscr{L}(T)\}$$

by truncating the exact product. Using the fast truncation algorithm of Lemma 2.10, *the complexity can be reduced to*

$$\begin{aligned} N_{\mathscr{H}, \odot}(T, k, k') \leq\ & 56 C_{\mathrm{sp}}^2 \max\{C_{\mathrm{id}}, C_{\mathrm{sp}}\} \max\{k, k'\}^2 (p + 1)^2 \#I \\ & + 184 C_{\mathrm{sp}} C_{\mathrm{id}} \max\{k, k'\}^3 (p + 1) \#\mathscr{L}(T). \end{aligned}$$

We call this mapping $\odot$ *or* $\odot^{\text{fast}}$ *in contrast to* $\odot^{\text{best}}$ *from above.*

Proof. a) (Rank) Due to (2.9), in each leaf of $T \cdot T$ the rank is bounded by $(p+1)C_{\text{sp}} \max\{k, k'\}$. If a leaf from T is contained in a leaf from $T \cdot T$, then the restriction to the leaf from T does not increase the rank. If a leaf from T contains leaves from $T \cdot T$ then their number is bounded by C_{id} and therefore the rank bounded by $\tilde{k}$.

b) (Complexity) We split the cost estimate into three parts: N_{mul} for calculating the exact product in $T \cdot T$, N_{-} for converting the $R(\tilde{k})$-blocks corresponding to "small" leaves $\mathscr{L}^{-}(T)$ in full matrix format and $N_{+}, N_{+}^{\text{fast}}$ for the (fast) truncation of the $R(\tilde{k})$-blocks to "large" leaves $\mathscr{L}^{+}(T)$ of rank k''.

b1) (N_{mul}) According to Theorem 2.20 and Lemma 2.4, the exact product using the $R(\tilde{k})$-representation in each leaf can be computed with complexity $4C_{\text{sp}}^3(p+1)^2 kk' \# I$.

b2) (N_{-}) In the "small" leaves $r \times s \in \mathscr{L}^{-}(T)$ we have to change the representation to full matrix format which has a cost of $2\tilde{k}\#r\#s$:

$$N_{-} \leq \sum_{r\times s \in \mathscr{L}^{-}(T)} 2\tilde{k}\#r\#s \;\leq \sum_{r\times s\in\mathscr{L}(T)} 2\tilde{k} n_{\min}(\#r + \#s) \;\leq\; \sum_{i=0}^{p} \sum_{r\times s\in\mathscr{L}(T,i)} 2\tilde{k} n_{\min}(\#r+\#s)$$

$$\overset{\text{Rem.1.10b}}{\leq} 4(p+1)C_{\text{sp}}\tilde{k} n_{\min}\#I \;\leq\; 4(p+1)^2 C_{\text{sp}}^2 C_{\text{id}} \max\{k,k'\} n_{\min}\#I.$$

b3) (N_{+}) For each "large" leaf in $\mathscr{L}^{+}(T)$ we truncate the $R(\tilde{k})$-block to rank k using Lemma 2.9 for the truncation or Lemma 2.10 for the fast truncation:

$$\begin{aligned}
N_{+} &\overset{\text{Lem.2.9}}{\leq} 6\tilde{k} N_{\mathscr{H},St}(T,\tilde{k}) + 23(\tilde{k})^3 \#\mathscr{L}(T)\\
&\overset{\text{Lem.2.4}}{\leq} 12 C_{\text{sp}}^3 C_{\text{id}}^2 \max\{k,k'\}^2 (p+1)^3 \#I + 23 C_{\text{sp}}^3 C_{\text{id}}^3 \max\{k,k'\}^3 (p+1)^3 \#\mathscr{L}(T)\\
&\leq\; 35 C_{\text{sp}}^3 C_{\text{id}}^3 \max\{k,k'\}^3 (p+1)^3 \max\{\#I, \#\mathscr{L}(T)\},\\
N_{+}^{\text{fast}} &\overset{\text{Lem.2.10}}{\leq} C_{\text{sp}} C_{\text{id}} (p+1)\big(24 \max\{k,k'\} N_{\mathscr{H},St}(T, \max\{k,k'\})\\
&\qquad + 184 \max\{k,k'\}^3 \#\mathscr{L}(T)\big)\\
&\overset{\text{Lem.2.4}}{\leq} 48 C_{\text{sp}}^2 C_{\text{id}} \max\{k,k'\}^2 (p+1)^2 \#I + 184 C_{\text{sp}} C_{\text{id}} \max\{k,k'\}^3 (p+1) \#\mathscr{L}(T).
\end{aligned}$$

2.2.2 General Case

Now we consider the general case of possibly different index sets I, J, K.

In Theorem 2.20 the cost for the exact multiplication $A \cdot B$ of two matrices from $\mathscr{H}(T,k)$ and $\mathscr{H}(T',k')$ is estimated and it turns out that the product lies in the set of $\mathscr{H}$-matrices based on the product tree $T \cdot T'$ (with increased rank). In practice, the structure in which the product has to be stored (after some kind of conversion) is given. If T is based on T_I, T_J and T' is based on T_J, T_K, then we assume that the target tree T'' is based on T_I, T_K. Consequently, each leaf of T'' is either

- contained in a leaf of $T \cdot T'$ or
- a vertex of $T \cdot T'$.

The following algorithm deals with the second case where the product of two structured matrices has to be computed and converted to $R(k'')$-format. To do this as fast as possible, we simultaneously compute the product of the two structured matrices and apply the hierarchical conversion of Algorithm 2.11.

Algorithm 2.25 (simultaneous multiplication and conversion to $R(k'')$ -format) *Let T be a block $\mathscr{H}$-tree based on T_I, T_J, let T' be a block $\mathscr{H}$-tree based on T_J, T_K and let T'' be a block $\mathscr{H}$-tree based on T_I, T_K. Let $A \in \mathscr{H}(T,k), B \in \mathscr{H}(T',k')$ be $\mathscr{H}$-matrices.*

First, we sketch the idea for a 2×2 *partitioning of the index set: assume we want to convert the product*

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

to $R(k'')$-format. By induction, we have already computed $R(k'')$-approximations to $A_{ij}B_{j\ell}$ for $i,j,\ell \in \{1,2\}$. The sum $A_{i1}B_{1\ell} + A_{i2}B_{2\ell}$ is then converted to an $R(k'')$-matrix $R_{i\ell}$. Therefore, we have to approximate the matrix consisting of the four $R(k'')$-submatrices $R_{11}, R_{12}, R_{21}, R_{22}$ by an $R(k'')$-matrix. This can be accomplished if we treat $\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & R_{12} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ R_{21} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & R_{22} \end{bmatrix}$ *as an $R(4k'')$-matrix and use the truncation $\mathscr{T}^{R}_{k'' \leftarrow 4k''}$ from Lemma* 1.3. *To compute the (formatted) product $C := A \odot B$ we use the following procedure called by "$C := 0$; $MulAddRk(C, I, J, K, A, B)$":*

procedure $MulAddRk(\text{var } C, r, s, t, A, B)$;

begin

  $C' := 0 \in \mathbb{R}^{r \times s}$.   if $S(r \times s) = \emptyset$ or $S(s \times t) = \emptyset$ then

  begin $C' := A|_{r \times s} B|_{s \times t}$; $\{$ *full or* $R(k) -$ *or* $R(k') -$ *matrix*$\}$

    $C := \mathscr{T}^{\mathscr{R}}_{k'' \leftarrow k'' + \max\{k,k',n_{\min}\}}(C + C')$

  end else

  begin for each $r' \in S(r), s' \in S(s), t' \in S(t)$ do

      $MulAddRk(C'|_{r' \times t'}, r', s', t', A, B)$;

    $C := \mathscr{T}^{\mathscr{R}}_{k'' \leftarrow \#S(r)\#S(t)k''}(C + C')$

end end;

At last we are able to present the algorithm for the fast $\mathscr{H}$-matrix multiplication $\odot$.

Algorithm 2.26 (fast $\mathscr{H}$-matrix multiplication) *Let T be a block $\mathscr{H}$-tree based on T_I, T_J, let T' be a block $\mathscr{H}$-tree based on T_J, T_K and let T'' be a block $\mathscr{H}$-tree based on T_I, T_K. Let $A \in \mathscr{H}(T,k)$ and $B \in \mathscr{H}(T',k')$ be $\mathscr{H}$-matrices. The following procedure computes a matrix $C \in \mathscr{H}(T'',k'')$ such that C approximates $A \cdot B$ by the fast truncation of Lemma* 2.10 *and Algorithm* 2.25.

$C = A \odot B$ is obtained by the call "$C := 0$; MulAdd(C, I, J, K, A, B)" *of*

procedure *MulAdd*(var C, r, s, t, A, B);

begin

  if $S(r \times s) \neq \emptyset$ and $S(s \times t) \neq \emptyset$ and $S(r \times t) \neq \emptyset$ then   { *all matrices subdivided* }

    for $r' \in S(r), s' \in S(s), t' \in S(t)$ do *MulAdd* (C, r', s', t', A, B)

  else if $S(r \times t) \neq \emptyset$ then   { *target matrix subdivided* }

    begin $C' := A|_{r \times s} B|_{s \times t}$;   { *full or* $R(k)-$ *or* $R(k')-$*matrix* }

      $C := \mathscr{T}^{\mathscr{H}}_{k'' \leftarrow k'' + \max\{k, k', n_{\min}\}}(C + C')$

    end else *MulAddRk* (C, r, s, t, A, B) { *target matrix not subdivided* }

end;

2.3 Inversion of $\mathscr{H}$-matrices

In order to explain the inversion procedure for $\mathscr{H}$-matrices, we will shortly recapitulate the idea of [8] for a quad tree T based on a binary tree T_I. Afterwards we introduce the (slightly more general) $\mathscr{H}$-matrix inversion algorithm and bound the complexity by the complexity of the matrix multiplication.

Example 2.27 (Inversion of a 2×2 block matrix) *Let $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ be a positive definite matrix. The inverse M^{-1} to M can be written in the form*

$$M^{-1} = \begin{bmatrix} (M_{11})^{-1}(I + M_{12}S^{-1}M_{21}(M_{11})^{-1}) & -(M_{11})^{-1}M_{12}S^{-1} \\ -S^{-1}M_{21}(M_{11})^{-1} & S^{-1} \end{bmatrix}, \; S := M_{22} - M_{21}(M_{11})^{-1}M_{12}. \tag{2.10}$$

The invertibility of M_{11} and S is ensured by the positive definiteness of M (*the supposed positive definiteness can be replaced by regularity of all principal submatrices*).

In (2.10) we use the multiplication and addition of matrices as well as the inverses $(M_{11})^{-1}$ and S^{-1}. The idea now is to replace the exact addition and multiplication by the formatted $\mathscr{H}$-matrix counterparts and define the two inverses in the subblocks recursively. This is done by the following algorithm.

Algorithm 2.28 ($\mathscr{H}$-matrix inversion) *The procedure Invert from Table 1 for the inversion of an $\mathscr{H}$-matrix M is to be called by "$H := 0$; $R := 0$; Invert(M, I, H, R);" where the inverse is returned in the matrix R, H is needed as auxiliary storage and the original matrix M is overwritten.*

Theorem 2.29 (complexity of the formatted inversion) *Let T be a block $\mathscr{H}$-tree. We assume that for the 'small' matrix blocks $r \times s \in \mathscr{L}^{-}(T)$ the complexity of the inversion is bounded by the complexity of the multiplication (in the case $n_{\min} = 1$ both are one elementary operation). Then the complexity $N_{\mathscr{H},\mathrm{Inv}}(T,k)$ of the formatted inversion (Algorithm 2.28) in the set $\mathscr{H}(T,k)$ is bounded by $N_{\mathscr{H},\odot}(T,k,k)$.*

Proof. We prove the statement by induction over the depth p of the tree T. For $p = 0$, we have assumed that the inversion is of the same complexity as the multiplication. Now let $p > 0$. For the inversion of the matrix we call the multiplication *MulAdd* for all combinations of blocks r_i, r_ℓ, r_j, where the combination $i = \ell = j$ stands for the inversion which is by induction at most of the same complexity as the multiplication. This is exactly what is done for the computation of the product of two $\mathscr{H}$-matrices. Additionally, we have to call $n-1$ times the formatted addition *Add* in the block $r_i \times r_j$, again the same for the product. □

Table 1. Procedure for the $\mathscr{H}$-matrix inversion

```
procedure Invert(var M, r, var H, var R);
begin if S(r×r) = ∅ then R|r×r := (M|r×r)^{-1} else                    {full submatrix}
  begin determine the sons S(r) = {r_1, ..., r_σ};      {elimination of the lower triangular blocks}
    for ℓ = 1, ..., σ do
    begin Invert(M, r_ℓ, H, R);
      for j = 1, ..., ℓ-1 do
      begin H|r_ℓ×r_j := 0; MulAdd(H|r_ℓ×r_j, r_ℓ, r_ℓ, r_j, R|r_ℓ×r_ℓ, R|r_ℓ×r_j); R|r_ℓ×r_j := H|r_ℓ×r_j end;
      for j = ℓ+1, ..., σ do
      begin H|r_ℓ×r_j := 0; MulAdd(H|r_ℓ×r_j, r_ℓ, r_ℓ, r_j, R|r_ℓ×r_ℓ, M|r_ℓ×r_j); M|r_ℓ×r_j := H|r_ℓ×r_j end;
      for i = ℓ+1, ..., σ do
      begin
        for j = 1, ..., ℓ do
        begin H|r_i×r_j := 0; MulAdd(H|r_i×r_j, r_i, r_ℓ, r_j, M|r_i×r_ℓ, R|r_ℓ×r_j);
              H|r_i×r_j := -H|r_i×r_j; Add(R|r_i×r_j, r_i, r_j, R|r_i×r_j, H|r_i×r_j)
        end;
        for j = ℓ+1, ..., σ do
        begin H|r_i×r_j := 0; MulAdd(H|r_i×r_j, r_i, r_ℓ, r_j, M|r_i×r_ℓ, R|r_ℓ×r_j);
              H|r_i×r_j := -H|r_i×r_j; Add(M|r_i×r_j, r_i, r_j, R|r_i×r_j, H|r_i×r_j)
    end end end;
    for ℓ = σ, ..., 1 do                                 {elimination of the upper triangular blocks}
      for i = ℓ-1, ..., 1 do
        for j = 1, ..., σ do
        begin H|r_i×r_j := 0; MulAdd(H|r_i×r_j, r_i, r_ℓ, r_j, M|r_i×r_ℓ, R|r_ℓ×r_j);
              H|r_i×r_j := -H|r_i×r_j; Add(R|r_i×r_j, r_i, r_j, H|r_i×r_j, H|r_i×r_j)
end end end;
```

3 Approximation of Matrices by $\mathscr{H}$-Matrices

In this section we first give an algebraic result concerning the structure of the inverse to an $\mathscr{H}$-matrix where the underlying tree T is the one from the right of the picture in Remark 2.21. Afterwards, we introduce the admissibility condition that is needed (in the applications that we aim for) to construct the tree T in such a way that the 'large' leaves $r \times s \in \mathscr{L}^+(T)$ allow for a low rank approximation of the matrix under consideration. In the context of partial differential equations it is the inverse to the stiffness or mass matrix that has to be stored (and computed), in the boundary element context it is the discrete operator that has to be stored (and computed).

For later purpose, we mention a result the proof of which is an easy exercise.

Lemma 3.1 *Let $M_i \in \mathscr{H}(T,k)$ converge to M as $i \to \infty$. Then also $M \in \mathscr{H}(T,k)$, i.e., $\mathscr{H}(T,k)$ is closed.*

3.1 Algebraic Approximation

In the practical applications, it is essential that although the inverse M^{-1} has a rather large local rank, we are able to approximate M^{-1} by a matrix from $\mathscr{H}(T,k)$ with modest rank k. In this subsection, however, we apply no truncation and show instead what the local rank of the exact inverse is.

Theorem 3.2 *Let $M \in \mathscr{H}(T,k)$ be an $\mathscr{H}$-matrix with minimal block size $n_{\min} = k$ and blockwise rank k. The block $\mathscr{H}$-tree T is based on a binary $\mathscr{H}$-tree T_I and for all $r \times s \in T$ we define*

$$S(r \times s) = \begin{cases} \{r' \times s' \mid r' \in S(r), s' \in S(s)\} & \text{if } r = s, \\ \emptyset & \text{otherwise} \end{cases}$$

(*similar to the block partitioning B_2 from* [8, *Section* 2.2.2]). *Let M be invertible and $p := \mathrm{depth}(T)$. Then the exact inverse M^{-1} to M fulfils*

$$M^{-1} \in \mathscr{H}(T, kp). \tag{3.1}$$

Proof. We prove the statement by induction over the depth p.

a) Start of induction ($p = 0$). The block $\mathscr{H}$-tree T consists only of the root $I \times I$. Since M was assumed to be of full rank it follows from Definition 1.14 that $\#I \le n_{\min}$. Furthermore, $\mathscr{L}(T) = \mathscr{L}^-(T) = \{I \times I\}$, $\mathscr{L}^+(T) = \emptyset$. Therefore $M^{-1} \in \mathscr{H}(T,0)$.

b) Induction step: let the statement be true for trees with depth $< p$. Let $S(I) = \{I_1, I_2\}$. The matrices M, M^{-1} are partitioned into 2×2 submatrices corresponding to the index sets I_1, I_2:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} (M^{-1})_{11} & (M^{-1})_{12} \\ (M^{-1})_{21} & (M^{-1})_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}. \tag{3.2}$$

By induction we have $(M_{11})^{-1} \in \mathcal{H}(T_{I_1 \times I_1}, k(p-1))$ and $(M_{22})^{-1} \in \mathcal{H}(T_{I_2 \times I_2}, k(p-1))$. Let us assume for a moment that M_{11} and M_{22} are invertible. Then equation (3.2) consists of four identities:

$$\begin{aligned}
(M^{-1})_{11} &= (M_{11})^{-1} - (M_{11})^{-1} M_{12} (M^{-1})_{21}, \\
(M^{-1})_{22} &= (M_{22})^{-1} - (M_{22})^{-1} M_{21} (M^{-1})_{12}, \\
(M^{-1})_{12} &= -(M_{11})^{-1} M_{12}, \\
(M^{-1})_{21} &= -(M_{22})^{-1} M_{21}.
\end{aligned}$$

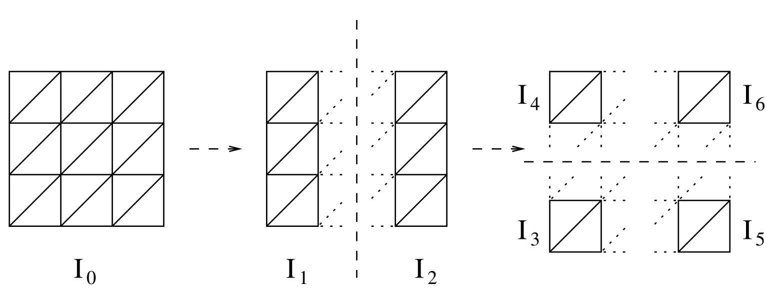
The matrices M_{12}, M_{21} are of rank at most k. The last two equations reveal that $\mathrm{rank}((M^{-1})_{12}) \le k \le kp$ and $\mathrm{rank}((M^{-1})_{21}) \le k \le kp$, the first two ones show $(M^{-1})_{11} \in \mathcal{H}(T_{I_1 \times I_1}, k(p-1)+k)$ and $(M^{-1})_{22} \in \mathcal{H}(T_{I_2 \times I_2}, k(p-1)+k)$.

c) If M_{11} or M_{22} is not invertible, then for any small enough $\varepsilon > 0$ the matrices $M_{11} + \varepsilon I$ and $M_{22} + \varepsilon I$ are invertible, so that $(M + \varepsilon I)^{-1} \in \mathcal{H}(T, kp)$. Application of Lemma 3.1 to the limit $\varepsilon \to 0$ yields (3.1). □

Example 3.3 (inversion of a special sparse matrix) *We consider a regular triangulation of* $[0,1]^2$ *with* $n^2 = 2^{2p}$, $p \in \mathbb{N}$, *degrees of freedom: the vertices of the grid are*

$$v_{ij} = \left(\frac{i-1}{n-1}, \frac{j-1}{n-1} \right), \qquad i, j = 1, \ldots, n.$$

Two vertices $v_{ij}, v_{i'j'}$ *are neighboured if* $|i - i'| + |j - j'| \le 1$ *or if* $i - i' = j - j'$ *and* $|i - i'| = 1$. *The index set is* $I := \{(i,j) \mid i, j = 1, \ldots, n\}$. *The index set* I *is divided successively as follows:*



In the first step we divide the index set I *into two equally sized subsets* $I_1 := \{(i,j) \mid i = 1, \ldots, n/2, j = 1, \ldots, n\}$ *and* $I_2 := \{(i,j) \mid i = n/2+1, \ldots, n, j = 1, \ldots, n\}$ *which are the two sons of the root* I. *In the second step we divide the index set* I_1 *into two equally sized subsets* $I_3 := \{(i,j) \mid i = 1, \ldots, n/2, j = 1, \ldots, n/2\}$ *and* $I_4 := \{(i,j) \mid i = 1, \ldots, n/2, j = n/2+1, \ldots, n\}$ *which are the two sons of* I_1, *analogously* I_2 *is divided into two sons* I_5, I_6. *We repeat steps one and two until the index subsets contain only one element: they are the leaves of the binary* $\mathscr{H}$*-tree* T_I.

The root of the block $\mathscr{H}$*-tree* T *is* $I \times I$. *The sons of a vertex* $r \times s \in T$ *are defined as required in Theorem* 3.2.

Let the matrix $M \in \mathbb{R}^{I \times I}$ *be sparse in the sense that* $M_{(i,j),(i',j')} = 0$ *if the corresponding vertices* $v_{ij}, v_{i'j'}$ *of the grid are not neighboured* (*This arises typically for finite element or finite difference discretisations of partial differential operators*).

The vertices of the $\mathscr{H}$*-tree* T_I *were chosen such that at most* n *elements of two disjoint index subsets* I_i, I_j *of* I *are neighboured. Therefore the rank* k *of* M *restricted to* $I_i \times I_j$ *is at most* n. *If* M *is invertible, then Theorem* 3.2 *yields*

$$M \in \mathscr{H}(T, n) \Rightarrow M^{-1} \in \mathscr{H}(T, np).$$

From [8] *we can estimate the storage requirements for the* $\mathscr{H}$*-matrix representation of* M^{-1} *by* $2n^3p^2$ *which* (*for* $p > 6$) *is less than* n^4 *for the full matrix representation.*

3.2 Analytic Approximation: Model Problem

We consider an integral operator of the form

$$\mathscr{K}[u](x) = \int_\Omega g(x,y)u(y)\mathrm{d}y$$

on a subdomain or submanifold $\Omega \subset \mathbb{R}^d$ with a kernel function $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. The operator $\mathscr{K}$ is discretised by a Galerkin finite element (boundary element) scheme for a basis $\mathscr{B} := \{\phi_1, \ldots, \phi_n\}$, $\phi_i : \Omega \to \mathbb{R}$, and yields a matrix

$$K_{i,j} := \int_\Omega \int_\Omega \phi_i(x) g(x,y) \phi_j(y) \mathrm{d}x\mathrm{d}y, \qquad i,j \in \{1, \ldots, n\}.$$

We denote the supports of the basis functions by

$$\Omega_i := \operatorname{supp} \phi_i \subset \Omega \qquad \text{and} \quad \Omega_\tau := \cup_{i \in \tau} \Omega_i \subset \Omega \text{ for } \tau \subset I.$$

Our aim is to approximate the matrix K by a matrix $K_{\mathscr{H}} \in \mathscr{H}(T, k)$ for a 'suitable' block $\mathscr{H}$-tree T and rank k. If one assumes that the kernel g is *asymptotically smooth* (cf. [2]) then it can locally be approximated by a degenerate kernel $\tilde{g}(x,y) = \sum_{i=1}^k g_{1,i}(x) g_{2,i}(y)$ such that

$$\max_{(x,y) \in \Omega_\tau \times \Omega_\sigma} |g(x,y) - \tilde{g}(x,y)| = \mathcal{O}(C_{\tau,\sigma}^{\sqrt[d]{k}})$$

for a block $\tau \times \sigma \in T_{I \times I}$, where the constant $C_{\tau,\sigma} < 1$ depends on the ratio of their distance $(\operatorname{dist}(\tau, \sigma) := \operatorname{dist}(\Omega_\tau, \Omega_\sigma))$ with respect to the Euclidean distance and their *Chebyshev diameter* (diam) defined by

$$\operatorname{diam}(\tau) := \inf\{\rho \in \mathbb{R} \mid \exists x \in \mathbb{R}^d \ \forall y \in \Omega_\tau : \ \|x - y\|_2 \le \rho/2\}. \tag{3.3}$$

Typically, one requires the *standard admissibility condition*

$$\min\{\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\} \le 2\eta \ \operatorname{dist}(\tau, \sigma) \tag{3.4}$$

to ensure $C_{\tau,\sigma} < 1$ (exponential convergence with respect to the rank k). However, the statements in this article also hold for the stronger admissibility condition

$$\max\{\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\} \le 2\eta \ \operatorname{dist}(\tau, \sigma) \tag{3.5}$$

(min replaced by max), which is needed for the (more refined) $\mathcal{H}^2$-matrix approach.

It is essential that the basis functions ϕ_i have a small support as usual in FEM or BEM. In the extreme opposite case of global support ($\Omega_i = \Omega$), there exists not even a single block $\tau \times \sigma$ that fulfils the admissibility condition (3.4). Therefore, we assume that the supports are locally separated in the sense that there exist two constants C_{sep} and $n_{\min}$ such that

$$\max_{i \in I} \#\{j \in I \mid \operatorname{dist}(\Omega_i, \Omega_j) \le C_{\text{sep}}^{-1} \operatorname{diam}(\Omega_i)\} \le n_{\min}. \tag{3.6}$$

The left-hand side is the maximal number of 'rather close' supports. Note that the bound $n_{\min}$ is the same as in Definition 2.3, i.e., the choice of $n_{\min}$ should satisfy (3.6). The constant C_{sep} is needed in the next section. The following example illustrates that C_{sep} is very small, even if the grid is strongly graded (note that the smaller C_{sep} is the weaker is the condition (3.6)).

Example 3.4 (geometrically graded mesh) *Let $\Omega = [0, 1]$ be an interval that is subdivided into n disjoint sub-intervals Ω_i:*

$$\Omega_n := [0, 2^{1-n}], \qquad \Omega_i := (2^{-i}, 2^{1-i}].$$

The mesh is geometrically graded to the left corner and fulfils condition (3.6) *for the constants $n_{\min} := 3$ and $C_{\text{sep}} := 3$ for any $n \in \mathbb{N}$ (the ratio of the diameters between two adjacent sub-intervals is $2 < C_{\text{sep}}$). A stronger grading would result in a larger C_{sep}.*

One should notice that an extremely refined mesh like in Example 3.4 is rarely used in practice.

Example 3.5 (algebraically graded mesh) *Usually, adaptive grids refined towards a point use an algebraically graded mesh like $\Omega_n := [0, n^{-g}]$, $\Omega_i := \left(\left(\frac{i-1}{n}\right)^g, \left(\frac{i}{n}\right)^g\right]$ for a suitable exponent $g \ge 1$ (see* [6]).

In our model problem we only consider the discretisation of an integral operator with sufficiently smooth kernel. However, the same admissibility condition (3.5) is also required to construct block $\mathscr{H}$-trees T that are suitable to approximate the inverse to a finite element stiffness matrix in the set $\mathscr{H}(T,k)$, where the underlying differential operator is uniformly elliptic with L^∞-coefficients (cf. [1]). Note that the integral kernel (the corresponding Green's function) has very poor smoothness, since the coefficients may be extremely nonsmooth.

4 Construction of the $\mathscr{H}$-Tree and Block $\mathscr{H}$-Tree

4.1 Construction of the $\mathscr{H}$-Tree T_I

Let I be any fixed (finite) index set. Let $d \in \mathbb{N}$. For each $i \in I$ we denote the Chebyshev centre x that yields the infimum in (3.3) for the support $\tau := \Omega_i$ of the basis function ϕ_i by m_i.

Construction 4.1 (cardinality balanced clustering) *Let $\{e_1, \dots, e_d\} \in \mathbb{R}^d$ denote the unit vectors. We construct the tree T_I by defining* $\mathrm{root}(T_I) := I$ *and for each vertex $t \in T$ the set $S(t)$ of successors as follows. We define the minimal and maximal coordinates*

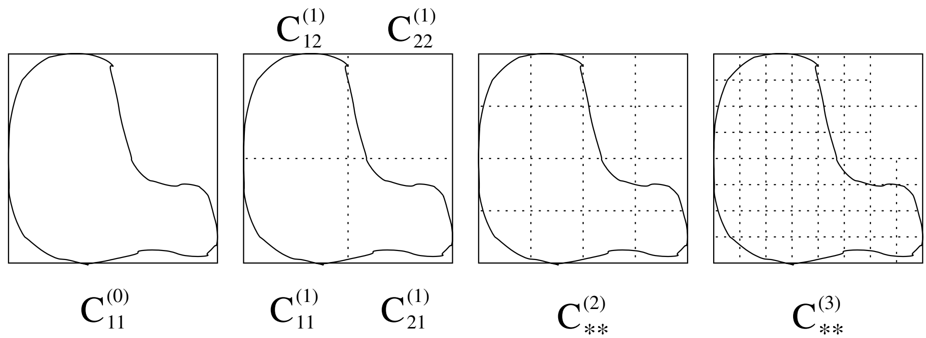
$$\alpha_j := \min\{\langle m_i, e_j\rangle \mid i \in t\}, \quad \beta_j := \max\{\langle m_i, e_j\rangle \mid i \in t\} \qquad \text{for} \quad j = 1, \dots, d.$$

Let $j_{\max} := \operatorname{argmax}\{\beta_j - \alpha_j \mid j \in \{1, \dots, d\}\}$. We sort the set $\{\langle m_i, e_{j_{\max}}\rangle \mid i \in t\}$ *in non-descending order $m_{i_1}, \dots, m_{i_{\#t}}$ (or determine the median). The set of sons of t is then defined as*

$$S(t) := \{s_1, s_2\}, \quad s_1 := \{i_1, \dots, i_{\lceil \#t/2 \rceil}\}, \quad s_2 := \{i_{\lceil \#t/2 \rceil + 1}, \dots, i_{\#t}\}.$$

The above defined cardinality balanced construction has shown to be practically useful. Later we will see that for some model problems we can prove that the cardinality balanced construction is well suited. In general however, we are not able to prove much for the resulting tree T_I, and therefore we give another easier to analyse procedure. In the numerical test of the last chapter we compare both approaches.

Construction 4.2 (geometrically balanced clustering) *Without loss of generality we assume that the domain Ω is contained in the cube $[0, h_{\max})^d$. The regular subdivision of this cube into $2^d, 2^{2d}, \dots, 2^{pd}$ subcubes can be used to define an $\mathscr{H}$-tree T_I with $\sum_{j=0}^{p} 2^{dj}$ vertices corresponding to one of the subcubes. We construct the tree T_I by defining* $\mathrm{root}(T_I) := I$ *and for each vertex $t \in T$ the set $S(t)$ of successors as follows. The cubes C_j^l on level l for a multiindex $j \in \mathbb{N}^d$ are defined as*



$$C^l_j := \mathscr{I}^l_{j_1} \times \cdots \times \mathscr{I}^l_{j_d} \qquad \textit{with} \quad \mathscr{I}^l_i := \left[(i-1)2^{-l}h_{\max}, i2^{-l}h_{\max}\right).$$

The sons (successors) $S(C^l_j)$ *are defined as the* 2^d *cubes on level* $l+1$ *that are contained in* C^l_j. *Each index subset* $t \in T_I^{(\ell)}$ *corresponds to a cube* C^l_j, *starting with the root I and the cube* $C^0_{(1,\dots,1)}$. *The sons of a vertex t with corresponding cube* C^l_j *are defined as*

$$S(t) := \{s_C \mid C \in S(C^l_j)\} \setminus \{\emptyset\}, \qquad \textit{where} \quad s_C := \{i \in t \mid m_i \in C\} \text{ for } C \in S(C^l_j).$$

4.2 Construction of the Block $\mathscr{H}$-Tree T

Based on the $\mathscr{H}$-tree T_I from Construction 4.1 or Construction 4.2 and the admissibility condition (3.4) we can define the block $\mathscr{H}$-tree T as follows. For an index subset $r \subset I$ we define the corresponding domain as $\Omega_r := \cup_{i\in r}\Omega_i$. A product index set $r \times s$ with corresponding cubes C_r and C_s is called admissible, if

$$\min\{\widetilde{\operatorname{diam}}(r), \widetilde{\operatorname{diam}}(s)\} \le \widetilde{\operatorname{dist}}(r,s), \tag{4.1}$$

where the modified distance and diameter are

$$\widetilde{\operatorname{diam}}(t) := \operatorname{diam}(C_t) + \max_{i\in t} \operatorname{diam}(\Omega_i),$$

$$\widetilde{\operatorname{dist}}(r,s) := \operatorname{dist}(C_r, C_s) - \max_{i\in r\cup s} \operatorname{diam}(\Omega_i).$$

If a product $r \times s$ is admissible with respect to (4.1) then (see Lemma 4.5) the corresponding domain $\Omega_r \times \Omega_s$ is admissible with respect to the standard admissibility condition (3.5).

Construction 4.3 (canonical block $\mathscr{H}$-tree) *Let the* $\mathscr{H}$*-tree* T_I *be given. We define the block* $\mathscr{H}$*-tree T by* $\operatorname{root}(T) := I \times I$ *and for each vertex* $r \times s \in T$ *the set of successors*

$$S(r \times s) := \begin{cases} \{r' \times s' | r' \in S(r), s' \in S(s)\} & \text{if } \#r > n_{\min} \text{ and } \#s > n_{\min} \\ & \quad \text{and } r \times s \text{ is in admissible,} \\ \emptyset & \text{otherwise.} \end{cases}$$

Lemma 4.4 *Let T be the block $\mathscr{H}$-tree of depth $p \geq 1$ built from the $\mathscr{H}$-tree T_I by Construction* 4.3. *We denote the maximal number of sons of a vertex $s \in T_I$ by* $\mathrm{C}_{\mathrm{sons}}$. *Then the sparsity constant* (*cf. Definition* 2.1) C_{sp} *of T is bounded by*

$$C_{\mathrm{sp}} \leq C_{\mathrm{sons}} \max_{r \in T_I} \#\{s \in T_I \mid r \times s \in T \setminus \mathscr{L}(T) \text{ and } r \times s \textit{ is } \text{inadmissible}\}.$$

Proof. Let $r \times s \in T^{(\ell)}$. Then $r \times s$ is either the root of T or the father element $\mathscr{F}(r) \times \mathscr{F}(s)$ is inadmissible due to Construction 4.3. □

Lemma 4.5 (geometrically balanced cluster tree) *Let* $h_{\min} := \min_{i \in I} \operatorname{diam}(\Omega_i)$. *We use the same notation as in Construction* 4.2 *and assume that* (3.6) *holds for some constants* $C_{\mathrm{sep}}, n_{\min} \in \mathbb{N}$. *Then Constructions* 4.2 *and* 4.3 *yield a block $\mathscr{H}$-tree T where each $r \times s \in \mathscr{L}^+(T)$ fulfils*

$$\min\{\operatorname{diam}(\Omega_r), \operatorname{diam}(\Omega_s)\} \leq 2\eta \operatorname{dist}(\Omega_r, \Omega_s)$$

and the depth as well as the sparsity and idempotency constant of T is bounded by

$$C_{\mathrm{sp}} \leq (8(\eta^{-1}(1 + C_{\mathrm{sep}}) + C_{\mathrm{sep}})\sqrt{d} + 4)^d,$$

$$C_{\mathrm{id}} \leq ((4 + 4\eta)(1 + C_{\mathrm{sep}}))^{2d},$$

$$\operatorname{depth}(T) \leq 1 + \log_2\Big((1 + C_{\mathrm{sep}})\sqrt{d} h_{\max} h_{\min}^{-1}\Big).$$

Proof. a) *Admissibility.* Let $r \times s \in \mathscr{L}^+(T)$ be admissible. Since $\operatorname{diam}(\Omega_r) \leq \widetilde{\operatorname{diam}}(r)$ and $\widetilde{\operatorname{dist}}(r, s) \leq \operatorname{dist}(\Omega_r, \Omega_s)$ we have

$$\widetilde{\operatorname{diam}}(r) \leq 2\eta \widetilde{\operatorname{dist}}\,(r, s) \Rightarrow \operatorname{diam}(\Omega_r) \leq 2\eta \operatorname{dist}(\Omega_r, \Omega_s).$$

b) *Sparsity*. For all $t \in T_I^{(\ell)}$ with $\#t > n_{\min}$ there holds

$$\max_{i \in t} \operatorname{diam}(\Omega_i) \overset{(3.6)}{\leq} C_{\mathrm{sep}} \operatorname{diam}(C_t) = C_{\mathrm{sep}} \sqrt{d} 2^{-\ell} h_{\max}, \tag{4.2a}$$

$$\widetilde{\operatorname{diam}}(t) = \quad \operatorname{diam}(C_t) + \max_{i \in t} \operatorname{diam}(\Omega_i) \leq (1 + C_{\mathrm{sep}})\sqrt{d} 2^{-\ell} h_{\max}, \tag{4.2b}$$

$$\widetilde{\operatorname{diam}}(t) \geq \quad \sqrt{d} 2^{-\ell} h_{\max}. \tag{4.2c}$$

Our aim is to apply Lemma 4.4 where we have to bound the number of inadmissible vertices. Let $r \in T_I$ with $\#r > n_{\min}$. The distance from C_r to the clusters belonging to the same level ℓ is considered in layers (see Fig. 1) as follows:

$$L_1 := \{C_s \mid \operatorname{dist}(C_s, C_r) = 0\}, \quad L_i := \{C_s \mid \operatorname{dist}(C_s, L_{i-1}) = 0\} \setminus L_{i-1} \quad \text{for } i = 2, 3, \ldots$$

The distance of a cluster Ω_s to Ω_r with $C_s \in L_{i+1}$ is bounded by

$$\widetilde{\operatorname{dist}}(s, r) \overset{(4.2a)}{\geq} (i - C_{\text{sep}}\sqrt{d})2^{-\ell}h_{\max}. \tag{4.2d}$$

For a cluster Ω_s with $C_s \in L_{i+1}$ there holds

$$\begin{aligned} i \geq (\eta^{-1} + \eta^{-1}C_{\text{sep}} + C_{\text{sep}})\sqrt{d} \quad &\Rightarrow \quad \eta(i - C_{\text{sep}}\sqrt{d}) \geq (1 + C_{\text{sep}})\sqrt{d} \\ &\Rightarrow \quad \eta(i - C_{\text{sep}}\sqrt{d})2^{-\ell}h_{\max} \geq (1 + C_{\text{sep}})\sqrt{d}2^{-\ell}h_{\max} \\ &\overset{(4.2b,d)}{\Rightarrow} \quad \min\{\widetilde{\operatorname{diam}}(s), \widetilde{\operatorname{diam}}(r)\} \leq \eta\,\widetilde{\operatorname{dist}}(s, r) \end{aligned}$$

and it follows that all products $s \times t$ with $C_s \in L_{i+1}$ and $i \geq i_{\text{layer}} := ((1 + C_{\text{sep}})\eta^{-1} + C_{\text{sep}})\sqrt{d}$ are admissible. The number of inadmissible clusters is therefore bounded by $\#L_1 + \cdots + \#L_{i_{\text{layer}}} \leq (2i_{\text{layer}} + 1)^d$. According to Lemma 4.4, the sparsity of T is bounded by $C_{\text{sp}} \leq 4^d(2i_{\text{layer}} + 1)^d$.

c) *Depth*. From (4.2b) the diameter of a (non-leaf) cluster on level ℓ is bounded by $(1 + C_{\text{sep}})\sqrt{d}2^{-\ell}h_{\max}$. According to the definition of $h_{\min}$ we get $[(1 + C_{\text{sep}})\sqrt{d}2^{-\ell}h_{\max} \geq h_{\min}$ and therefore $\ell \leq \log_2((1 + C_{\text{sep}})\sqrt{d}(h_{\max}/h_{\min}))$.

d) *Idempotency*. Let $r \times t \in \mathscr{L}(T, \ell)$. If $\#I_r \leq n_{\min}$ or $\#I_t \leq n_{\min}$, then the elementwise idempotency is $C_{\text{id}}(r \times t) = 1$. Now let $r \times t$ be admissible. Define $q := \lceil(\log_2(2(1 + \eta)(1 + C_{\text{sep}}))\rceil$. We want to prove that for all vertices $r', s', t' \in T^{\ell+q}$, $r' \times s' \in S^*(r \times s)$ and $s' \times t' \in S^*(s \times t)$ one of the vertices $r' \times s'$ and $s' \times t'$ is a leaf. Let r', s', t' be given as above and $\min\{\#r', \#s', \#t'\} > n_{\min}$.

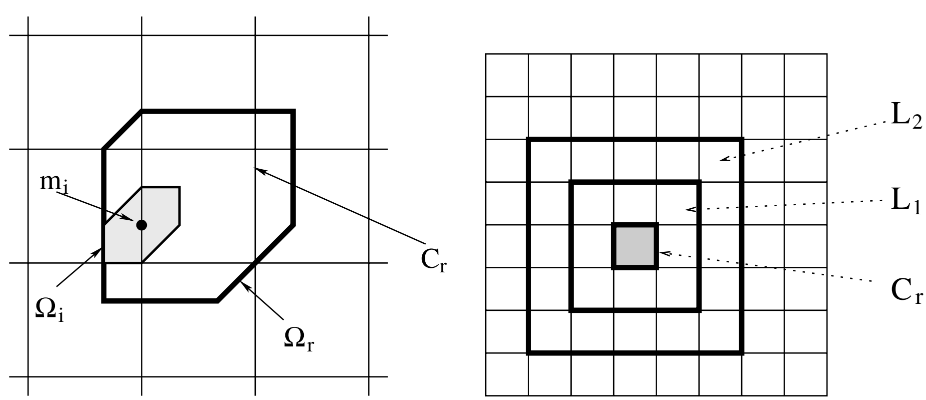


Fig 1. Left: the cluster Ω_r contains the whole set Ω_i while the corresponding cube C_r contains only m_i. Right: the cube C_r and the first two layers L_1 and L_2

For $u \in \{r', s', t'\}$ it holds

$$\widetilde{\operatorname{diam}}(u) \overset{(4.2b)}{\leq} (1 + C_{\text{sep}})\sqrt{d}2^{-q-\ell}h_{\max} \leq \frac{1}{2}\sqrt{d}(\eta+1)^{-1}2^{-\ell}h_{\max}. \tag{4.2e}$$

Then we can estimate

$$\begin{aligned}
\widetilde{\operatorname{diam}}(s') &\overset{(4.2e)}{\leq} \frac{1}{2}\sqrt{d}(1-\eta(\eta+1)^{-1})2^{-\ell}h_{\max} = \frac{1}{2}\sqrt{d}2^{-\ell}h_{\max} - \eta\frac{1}{2}\sqrt{d}2^{-\ell}h_{\max}(\eta+1)^{-1} \\
&\overset{(4.2c,e)}{\leq} \frac{1}{2}\min\{\widetilde{\operatorname{diam}}(r), \widetilde{\operatorname{diam}}(t)\} - \eta \max_{u\in\{r',s',t'\}} \widetilde{\operatorname{diam}}(u) \\
&\leq \frac{1}{2}\eta\,\widetilde{\operatorname{dist}}(r,t) - \eta \max_{u\in\{r',s',t'\}} \widetilde{\operatorname{diam}}(u) \\
&= \frac{1}{2}\eta\operatorname{dist}(C_r, C_t) - \frac{1}{2}\eta\max_{i\in r\cup t}\operatorname{diam}(\Omega_i) - \eta \max_{u\in\{r',s',t'\}} \widetilde{\operatorname{diam}}(u) \\
&\leq \eta\max\{\operatorname{dist}(C_{r'}, C_{s'}), \operatorname{dist}(C_{s'}, C_{t'})\} + \eta\operatorname{diam}(C_{s'}) - \eta \max_{u\in\{r',s',t'\}} \widetilde{\operatorname{diam}}(u) \\
&\leq \eta\max\{\widetilde{\operatorname{dist}}(r', s'), \widetilde{\operatorname{dist}}(s', t')\} + \eta \max_{i\in r'\cup s'\cup t'}\operatorname{diam}(\Omega_i) + \eta\operatorname{diam}(C_{s'}) \\
&\quad - \eta \max_{u\in\{r',s',t'\}} \widetilde{\operatorname{diam}}(u) \\
&\leq \eta\max\{\widetilde{\operatorname{dist}}(r', s'), \widetilde{\operatorname{dist}}(s', t')\},
\end{aligned}$$

i.e., either $r' \times s'$ or $s' \times t'$ is admissible (and has no sons). It follows that there are no vertices $r'' \times s'' \in T^{(\ell+q+1)}$ and $s'' \times t'' \in T^{(\ell+q+1)}$ with $r'' \in S^*(r), t'' \in S^*(t)$. Since the number of sons of a vertex is limited by 2^{2d}, there are at most 2^{2dq} vertices in $T \cdot T$ that are contained in $r \times t$. □

Remark 4.6 *Lemma* 4.5 *proves that Construction* 4.2 ($\rightarrow$ *$\mathcal{H}$-tree*) *combined with Construction* 4.3 ($\rightarrow$ *block $\mathcal{H}$-tree*) *yields an $\mathcal{H}$-tree T that is sparse and idempotent with C_{sp} and C_{id} independent of the cardinality of the index set I. The depth of the tree is estimated by the logarithm of the ratio of the smallest element to the diameter of the whole domain* (*which can be large*). *Construction* 4.1 *does not necessarily lead to sparsity* (*idempotency*) *independent of $\#I$. This is not to say that the resulting $\mathcal{H}$-matrices are not data-sparse, but the block-structure is less homogenous and more difficult to analyse. The trees from Construction* 4.1 *fulfil the condition $\#S(t) \neq 1$ for all vertices $t \in T$.*

Remark 4.7 *(admissibility for $\mathcal{H}^2$-matrices) The results of Lemma 4.5 depend on the admissibility condition (3.4). In the context of $\mathcal{H}^2$-matrices [11] the stronger admissibility condition*

$$\max\{\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\} \leq 2\eta \operatorname{dist}(\tau, \sigma) \tag{4.3}$$

is required. The bounds for the sparsity constant C_{sp}, *the idempotency constant* C_{id} *and the depth* p *of the tree also hold for the admissibility condition* (4.3), *because the reference cubes* C_r,C_s *on the same level are all of equal size.*

4.3 A Special Vertex Concentrated Grid

In Lemma 4.2 we were able to prove that the block $\mathscr{H}$-tree constructed by geometrically balanced clustering is sparse and almost idempotent with constants independent of the number of basis functions $\#I$. However, the depth p of the tree depends on the ratio of the diameter $h_{\min}$ of the smallest support to the diameter $h_{\max}$ of the whole domain. For a uniformly refined grid with n^d vertices in $\mathbb{R}^d$ we would expect $h_{\min} = \mathcal{O}(n^{-1})$, while $h_{\max} = \mathcal{O}(1)$. If the grid is concentrated along one edge using an algebraically graded mesh, we would expect $h_{\min} = \mathcal{O}(n^{-d})$ (see Example 3.5). In both cases the depth p is proportional to $\log(n)$. Therefore, for practically relevant grid constructions, the depth of the tree causes no problems.

However, there are pathological cases of geometrically graded meshes. In Example 3.4 the grid is exponentially concentrated towards the origin. The diameter of the leftmost interval is 2^{1-n} and the diameter of the domain is 1. Here, the depth of the (geometrically balanced) tree would be $p = \mathcal{O}(n)$ (implying that the $\mathscr{H}$-matrix technique is as costly of the naive approach, e.g., the storage is $N_{\mathscr{H},St} \leq \operatorname{depth}(T) C_{\text{sp}} \#I = \mathcal{O}(n^2)$).

In the following we consider a similar example in $\mathbb{R}^2$ where we can prove that the (almost) *cardinality balanced* $\mathscr{H}$-tree of the index set has sparsity and idempotency constants independent of n. In the next example, the elements (panels) may be considered as the supports of piecewise constant basis functions in a boundary element method.

Example 4.8 *We consider the grid from Fig.* 2 *with* $n = 3p + 1$ *panels* $(\Omega_i)_{i \in I}$ *that is constructed by* p *times regularly refining the panel at the origin into four parts and starting with the unit square* $[0, 1]^2$. *We define the layers* $L_1, \ldots, L_p$ *that contain panels of equal size* (*see Fig.* 2) *where the size is decreasing with increasing layer number.*

Let $J := \{1, \ldots, p\}$ *denote the layer numbers and let* T_J *be a cardinality balanced binary* $\mathscr{H}$*-tree of* J *as depicted in Fig.* 2. *Analogously, the tree* T_I *is the same as* T_J *but the layer numbers are replaced by the numbers of the domains that belong to it. The tree* T *is built as in Construction* 4.3 *with the admissibility condition*

$$r \times s \text{ admissible} \quad \Leftrightarrow \quad \min\{\operatorname{diam}(\Omega_r), \operatorname{diam}(\Omega_s)\} \leq 2\eta \operatorname{dist}(\Omega_r, \Omega_s)$$

and $n_{\min} := 2\lceil 3/2 + \log_2(\eta^{-1}) \rceil$. *In the following we bound the sparsity* C_{sp} *and idempotency* C_{id} *of* T.

The diameter of a single layer L_j *is* $2^{3/2-j}$ *which is also the diameter of* $L_j \cup \cdots \cup L_p$. *Two vertices* $r, s \in T_I$ *are admissible, if there are at least* $\lceil 3/2 + \log_2(\eta^{-1}) \rceil$ *layers*

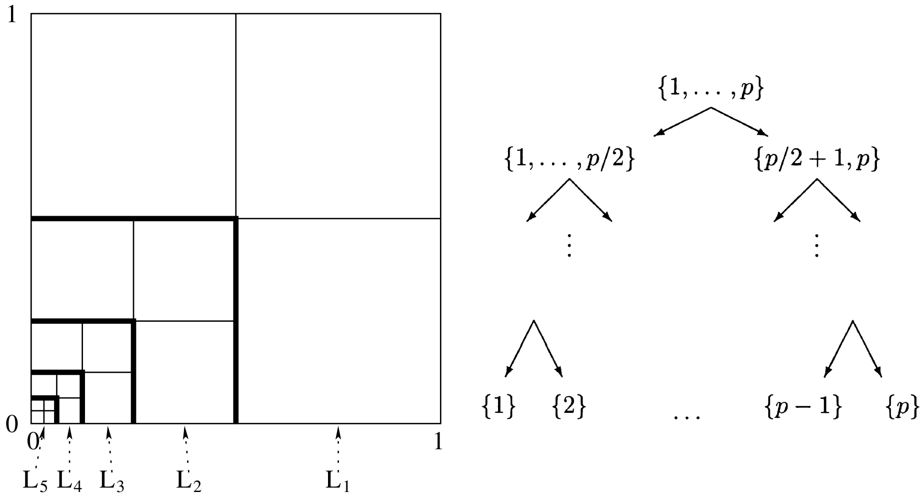


Fig. 2. Left: the grid consists of $n = 3p + 1$ panels. The diameter of the smallest panel is $2^{-p}\sqrt{2}$. The grid is partitioned into layers $L_1, \ldots, L_p$ that contain panels of equal size. Right: a balanced $\mathscr{H}$-tree for the index set $J = \{1, \ldots, p\}$, where p is a power of two. If p is not a power of two, then there appear also leaves on the last but one level

between them: *the smaller one is of the size* $2^{3/2-j}$ *and the distance between the two is at least* $2^{3/2+\log_2(\eta^{-1})}2^{-j} = \eta^{-1}2^{3/2-j}$.

(*Sparsity*) *Let* $r \in T_I$. *According to the prior statement the only inadmissible nodes to r are the ones containing at least one of the* $2\lceil 3/2 + \log_2(\eta^{-1})\rceil + 1$ *layers closest to r. From Lemma* 4.4 *we get* $C_{\text{sp}} \leq 8 + 4\lceil \log_2(\eta^{-1})\rceil$.

(*Idempotency*) *Let* $r \times t \in T$ *be admissible and let* $r \times s$, $s \times t$ *be inadmissible*, *especially* $\#s > n_{\min}$. *Then s contains at least* $2\lceil 3/2 + \log_2(\eta^{-1})\rceil$ *layers so that the two sons of s have at least a distance of* $\lceil 3/2 + \log_2(\eta^{-1})\rceil$ *to one of the clusters* r, t. *Therefore*, $C_{\text{id}} \leq \#S(r)\#S(t) + 1 \leq 5$.

In the previous example we were able to define the $\mathscr{H}$-tree T_I such that the canonical block $\mathscr{H}$-tree T from Construction 4.3 is sparse and idempotent with $\text{depth}(T) = \mathcal{O}(\log(n))$. This example illustrates that in the case where the geometrically balanced approach of Construction 4.2 fails, we can use Construction 4.1 which will yield an $\mathscr{H}$-tree T_I of depth at most $\log_2(n)$. In practice however, we do not expect the grids to be strongly refined only towards a few single vertices and therefore Construction 4.2 should be appropriate.

5 Numerical Results

The numerical tests in this section serve two purposes: first, we want to compare the theoretical results with the numerical ones in order to see if there is some gap between theoretical asymptotic bounds and actual complexity. Second, for the cardinality balanced clustering we were not able to sufficiently analyse the arising

$\mathscr{H}$-trees and therefore we want to observe the complexity of the $\mathscr{H}$-matrix arithmetics for some model problems. It will turn out that the complexity is geometry independend in the sense that it is worse for a uniform grid than for irregular grids.

It should be noted that the operator to be inverted has no influence on the complexity of the formatted arithmetics for a fixed rank k (only the approximation quality may differ). For the sake of simplicity, we consider Poisson's equation in the next subsections. Numerical results for other operators are presented in the last Subsection 5, see also [12].

All computations in this chapter were performed on a SUN ULTRASPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate.

5.1 Model Problem

We consider Poisson's equation

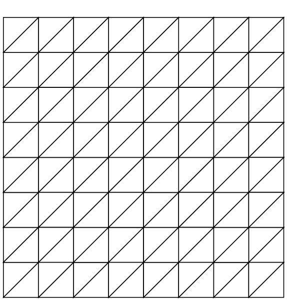
$$-\Delta u = f$$

in $\Omega \subset \mathbb{R}^2$ with Dirichlet boundary conditions $u|_\Gamma = 0$ on $\Gamma := \partial\Omega$. A Ritz-Galerkin discretisation with basis functions $(\phi_i)_{i=1}^n$ leads to the problem of solving a linear system of equations

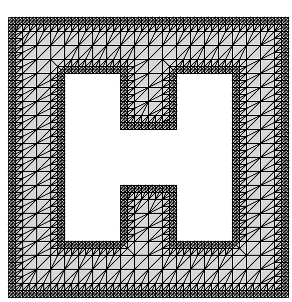
$$Ax = b$$

for the right hand side $b \in \mathbb{R}^n$ with entries $b_i := \int_\Omega f(x)\phi_i(x)\mathrm{d}x$ and the stiffness matrix $A \in \mathbb{R}^{n\times n}$ with entries $A_{i,j} := \int_\Omega \int_\Omega (\nabla\phi_j(x))^T \nabla\phi_i(y)\mathrm{d}x\mathrm{d}y$. We choose the nodal basis for the piecewise linear functions on a triangulation of the domain Ω.

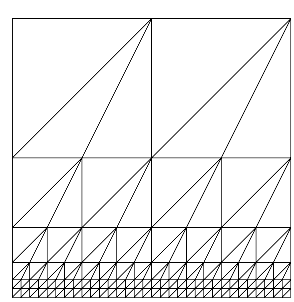
Our goal is to compute and store an approximation $\widetilde{A^{-1}}$ to A^{-1} in the $\mathscr{H}$-matrix format. For the domain and triangulation we consider the three cases of a regular refinement of the unit square, a boundary concentrated grid and an edge concentrated grid. The variety of triangulations is used to compare the clustering algorithms. It turns out that the uniform grid is the worst case with respect to the complexity of the (formatted) arithmetics per degree of freedom. Therefore, the numerical results for the uniform triangulation can be regarded as a benchmark result for arbitrary triangulations.



uniform grid



boundary concentrated



edge concentrated

5.2 Uniform Grid

The $\mathscr{H}$-tree T_I^{card} built by Construction 4.1 (cardinality balanced) and the $\mathscr{H}$-tree T_I^{geo} built by Construction 4.2 (geometrically balanced) for the index set $I := \{1, \ldots, n\}$ coincide in the uniform case. Construction 4.3 yields the block $\mathscr{H}$-tree T whose leaves partition the product index set $I \times I$. The parameter η in the admissibility condition is $\eta := 1.0$ and the minimal blocksize is $n_{\min} := 32$. For $n = 1024$ and $n = 4096$ degrees of freedom the partitioning is depicted in Fig. 3. The sparsity C_{sp} and the idempotency C_{id} of the tree T are given in Table 2. We observe that the sparsity is bounded by 23 and the idempotency is bounded by 18. The complexity of the (formatted) arithmetics can be seen in Tables 3–5. The estimated complexity for the (formatted) multiplication and the (formatted) inversion is due to Theorem 2.24 and 2.29 $\mathcal{O}(n \log(n)^2 k^2)$. For a fixed rank k and an increase of the number of degrees of freedom from $n = 65536 = 2^{16}$ to $4n = 262144 = 2^{18}$ we expect an increase in the complexity by a factor of $\frac{4n \log(4n)^2}{n \log(n)^2} = 81/16 \approx 5$. This happens for the (formatted) multiplication in Table 4 for $k = 1$ and the (formatted) inversion in Table 3 for $k \in \{1, 2\}$. The (formatted) inversion is by a factor of $2 - 3$ faster than the (formatted) multiplication, because the sparsity of the stiffness matrix is exploited in the computational scheme.

5.3 Boundary Concentrated Grid

The $\mathscr{H}$-tree T_I^{card} built by Construction 4.1 (cardinality balanced) and the $\mathscr{H}$-tree T_I^{geo} built by Construction 4.2 (geometrically balanced) for the index set

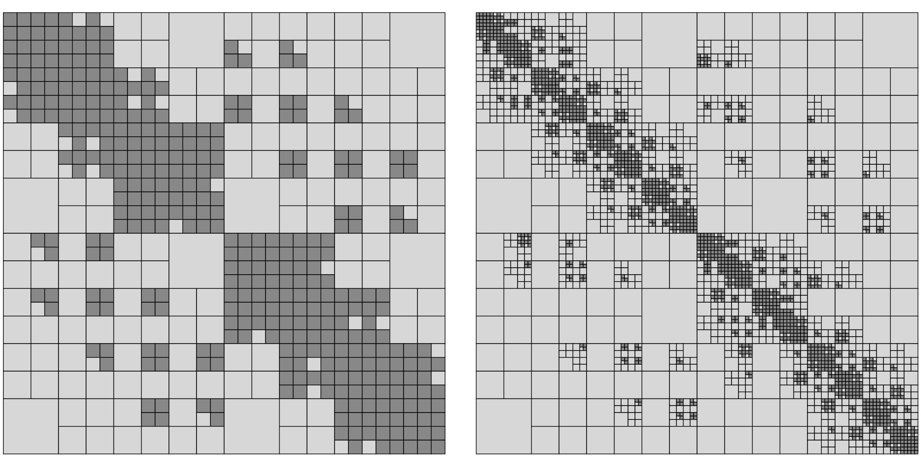


Fig. 3. The partitioning of the product index set $I \times I$ in the uniform case for $n = 1024$ and $n = 4096$ degrees of freedom. $R(k)$-blocks are light grey and full matrix blocks are dark grey

Table 2. The sparsity C_{sp} and the idempotency C_{id} of the tree T is bounded for increasing n

	$n = 4096$	$n = 16384$	$n = 65536$	$n = 262144$
C_{sp}	23	23	23	23
C_{id}	18	18	18	18

Table 3. Left: time (in seconds) for the (formatted) inversion on a uniform grid. Right: relative error $\|I - A\ \mathrm{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on a uniform grid

$n =$	4096	16384	65536	262144
$k = 1$	10.59	6.7+1	3.5+2	1.6+3
$k = 2$	11.85	8.0+1	4.4+2	2.2+3
$k = 3$	13.73	1.0+2	5.6+2	3.0+3
$k = 4$	16.19	1.2+2	6.8+2	3.6+3
$k = 5$	19.33	1.5+2	8.6+2	4.8+3
$k = 6$	22.41	1.7+2	1.0+3	6.0+3
$k = 7$	25.80	2.0+2	1.3+3	7.4+3
$k = 8$	27.87	2.2+2	1.3+3	7.8+3
$k = 9$	30.19	2.4+2	1.5+3	9.1+3
$k = 15$	39.77	3.4+2	2.3+3	1.5+4
$k = 20$	42.15	3.7+2	2.6+3	1.6+4

$n =$	4096	16384	65536	262144
$k = 1$	2.4	8.9	2.6+1	4.7+1
$k = 2$	5.7−1	3.2	1.2+1	2.7+1
$k = 3$	9.2−2	5.2−1	2.4	1.0+1
$k = 4$	2.0−2	9.9−2	4.4−1	1.91
$k = 5$	2.3−3	9.2−3	4.0−2	1.7−1
$k = 6$	6.4−4	3.7−3	1.8−2	8.4−2
$k = 7$	1.4−4	6.9−4	2.9−3	1.2−2
$k = 8$	7.8−5	3.9−4	1.8−3	7.7−3
$k = 9$	8.5−6	4.6−5	2.1−4	9.4−4
$k = 15$	6.8−9	3.3−8	1.3−7	5.2−7
$k = 20$	1.7−12	1.3−10	5.3−10	2.5−9

Table 4. Left: time (in seconds) for the (formatted) addition on a uniform grid. Right: time (in seconds) for the (formatted) multiplication on a uniform grid

$n =$	4096	16384	65536	262144
$k = 1$	1.5−1	0.81	4.1	2.0+1
$k = 2$	2.4−1	1.44	7.8	4.0+1
$k = 3$	3.4−1	2.09	1.2+1	6.2+1
$k = 4$	4.9−1	2.87	1.6+1	8.0+1
$k = 5$	6.7−1	4.03	2.2+1	1.1+2
$k = 6$	8.9−1	5.27	2.8+1	1.5+2
$k = 7$	1.12	6.82	3.7+1	1.9+2
$k = 8$	1.33	8.02	4.3+1	2.2+2
$k = 9$	1.66	10.19	5.5+1	2.9+2
$k = 15$	3.94	24.21	1.3+2	6.9+2
$k = 20$	5.08	34.04	1.9+2	1.0+3

$n =$	4096	16384	65536	262144
$k = 1$	20.59	1.3+2	6.6+2	3.3+3
$k = 2$	24.95	1.7+2	9.4+2	5.2+3
$k = 3$	30.36	2.1+2	1.3+3	7.6+3
$k = 4$	37.82	2.7+2	1.6+3	9.3+3
$k = 5$	46.91	3.5+2	2.2+3	1.3+4
$k = 6$	57.02	4.3+2	2.8+3	1.7+4
$k = 7$	68.75	5.3+2	3.5+3	2.2+4
$k = 8$	77.28	6.0+2	3.8+3	2.3+4
$k = 9$	93.55	7.3+2	4.8+3	2.9+4
$k = 15$	1.7+2	1.4+3	9.7+3	6.2+4
$k = 20$	2.0+2	1.8+3	1.2+4	8.0+4

Table 5. Left: time (in seconds) for the matrix vector multiplication on a uniform grid. Right: storage requirements (in 1024 Byte) for an $\mathcal{H}$-matrix corresponding to a uniform grid

$n =$	4096	16384	65536	262144
$k = 1$	2.0−2	0.16	0.76	3.3
$k = 2$	3.4−2	0.18	0.88	3.9
$k = 3$	3.7−2	0.20	0.99	4.6
$k = 4$	4.0−2	0.22	1.11	5.2
$k = 5$	4.3−2	0.24	1.23	5.9
$k = 6$	4.6−2	0.26	1.35	6.5
$k = 7$	4.9−2	0.28	1.46	7.1
$k = 8$	5.1−2	0.30	1.58	7.7
$k = 9$	5.4−2	0.31	1.69	8.4
$k = 15$	7.1−2	0.43	2.39	12.1
$k = 20$	7.1−2	0.44	2.54	13.4

$n =$	4096	16384	65536	262144
$k = 1$	1.5+4	7.4+4	3.3+5	1.4+6
$k = 2$	1.7+4	8.6+4	4.0+5	1.8+6
$k = 3$	1.9+4	9.8+4	4.7+5	2.1+6
$k = 4$	2.1+4	1.1+5	5.4+5	2.5+6
$k = 5$	2.2+4	1.2+5	6.1+5	2.9+6
$k = 6$	2.4+4	1.3+5	6.8+5	3.3+6
$k = 7$	2.6+4	1.5+5	7.5+5	3.6+6
$k = 8$	2.7+4	1.6+5	8.2+5	4.0+6
$k = 9$	2.9+4	1.7+5	8.8+5	4.4+6
$k = 15$	3.9+4	2.4+5	1.3+6	6.7+6
$k = 20$	4.8+4	3.0+5	1.7+6	8.5+6

$I := \{1, \ldots, n\}$ differ in the boundary concentrated case. Construction 4.3 yields the block $\mathcal{H}$-tree T^{card} or T^{geo}, respectively, whose leaves partition the product index set $I \times I$. The parameter η in the admissibility condition is $\eta := 1.0$ and the minimal blocksize is $n_{\min} := 32$. For $n = 3058$ degrees of freedom the partitioning (geometrically and cardinality balanced) is depicted in Fig. 4. The sparsity C_{sp} and the idempotency $C_{\rm id}$ of the trees $T^{\rm card}$ and $T^{\rm geo}$ are given in Table 6. As we expected the sparsity and idempotency constants of the tree $T^{\rm geo}$ are bounded while these values seem to increase for the tree $T^{\rm card}$. The complexity and accuracy of the (formatted) inversion is given in Table 7. The complexity of the inversion is reduced as compared to the uniform case while the accuracy is enhanced. This resembles the fact that the grid degenerates to a lower dimensional structure (the boundary). The cardinality balanced tree $T^{\rm card}$ is also suitable for the (formatted) arithmetics, although it is of an irregular structure and does not possess a bounded sparsity or idempotency. From Table 8 we observe that the complexity of the corresponding arithmetic operations exceeds that of the geometrically balanced tree by a factor of $2-3$. Since we were able to provide estimates for the complexity with respect to the geometrically balanced tree and this tree yields a better performance in practice, we propose to use the tree $T^{\rm geo}$ over the tree $T^{\rm card}$.

5.4 Edge Concentrated Grid

The $\mathscr{H}$-tree $T_I^{\rm card}$ built by Construction 4.1 (cardinality balanced) and the $\mathscr{H}$-tree $T_I^{\rm geo}$ built by Construction 4.2 (geometrically balanced) for the index set

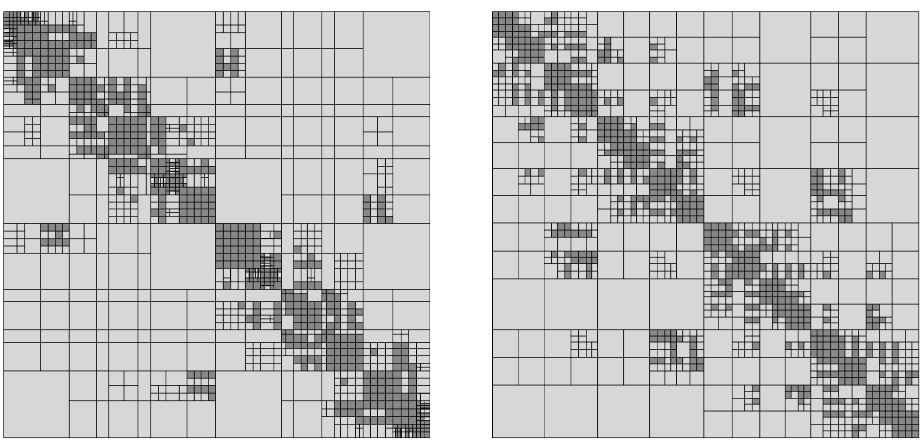


Fig. 4. The partitioning of the product index set $I \times I$ in the boundary concentrated case for $n = 3216$ degrees of freedom; to the left the geometrically balanced and to the right the cardinality balanced case. $R(k)$-blocks are light grey and full matrix blocks are dark grey

Table 6. The sparsity $C_{\rm sp}$ and the idempotency $C_{\rm id}$ of the tree $T^{\rm geo}$ are bounded for increasing n while this is not true for $T^{\rm card}$ in the boundary concentrated case

$n =$	6664	13568	27384	55024	110312
depth$(T^{\rm geo})$	11	13	15	17	19
$C_{\rm sp}(T^{\rm geo})$	26	28	34	36	26
$C_{\rm id}(T^{\rm geo})$	18	20	24	22	20
depth$(T^{\rm card})$	9	10	11	12	13
$C_{\rm sp}(T^{\rm card})$	32	38	56	80	131
$C_{\rm id}(T^{\rm card})$	24	28	34	39	53

Table 7. Left: time (in seconds) for the (formatted) inversion on the boundary concentrated grid for the geometrically balanced tree T^{geo}. Right: relative error $\|I - A\,\text{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on the boundary concentrated grid

$n =$	6664	13568	27384	55024	110312
$k = 1$	17.83	30.23	7.5+1	1.8+2	4.0+2
$k = 2$	19.24	33.72	8.6+1	2.1+2	4.7+2
$k = 3$	21.27	38.26	9.6+1	2.4+2	5.5+2
$k = 4$	22.79	43.09	1.0+2	2.5+2	5.7+2
$k = 5$	24.53	44.47	1.1+2	2.7+2	6.3+2
$k = 6$	25.03	46.66	1.2+2	2.9+2	6.7+2
$k = 7$	26.42	47.88	1.2+2	3.0+2	7.0+2
$k = 8$	25.71	47.81	1.2+2	2.9+2	6.9+2
$k = 9$	25.72	47.94	1.2+2	3.0+2	7.0+2

$n =$	6664	13568	27384	55024	110312
$k = 1$	9.6−2	9.9−2	7.9−2	1.1−1	9.4−2
$k = 2$	1.3−2	1.1−2	1.7−2	1.9−2	1.6−2
$k = 3$	3.9−3	4.4−3	1.7−3	4.5−3	4.7−3
$k = 4$	8.6−5	4.7−4	1.7−4	5.0−4	5.1−4
$k = 5$	8.9−6	3.6−5	7.6−6	4.9−5	5.0−5
$k = 6$	2.1−8	9.8−7	1.2−6	1.3−6	1.4−6
$k = 7$	3.1−10	5.0−7	1.9−10	5.8−7	5.9−7
$k = 8$	1.4−12	4.2−10	2.1−11	2.5−10	2.8−10
$k = 9$	1.0−14	2.4−13	2.1−14	2.7−13	2.8−13

Table 8. Left: time (in seconds) for the (formatted) inversion on the boundary concentrated grid for the cardinality balanced tree T^{card}. Right: relative error $\|I - A\,\text{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on the boundary concentrated grid

$n =$	6664	13568	27384	55024	110312
$k = 1$	25.69	68.26	1.9+2	4.1+2	8.2+2
$k = 2$	27.08	73.67	2.1+2	4.6+2	9.7+2
$k = 3$	29.29	80.22	2.3+2	5.4+2	1.2+3
$k = 4$	31.07	87.48	2.5+2	5.9+2	1.3+3

$n =$	6664	13568	27384	55024	110312
$k = 1$	3.8−2	7.8−2	7.8−2	1.0−1	1.1−1
$k = 2$	4.8−3	2.2−2	2.2−2	2.7−2	2.8−2
$k = 3$	1.2−3	6.3−3	6.3−3	8.5−3	8.6−3
$k = 4$	1.2−4	3.9−4	3.9−4	1.1−3	1.2−3

Table 9. The sparsity C_{sp} and the idempotency C_{id} of the tree T^{geo} are bounded for increasing n while this is not true for T^{card} in the edge concentrated case

$n =$	6129	12272	24559	49134	98285
$\text{depth}(T^{\text{geo}})$	13	15	17	19	21
$C_{\text{sp}}(T^{\text{geo}})$	21	21	21	21	21
$C_{\text{id}}(T^{\text{geo}})$	16	16	16	16	16
$\text{depth}(T^{\text{card}})$	10	11	12	13	14
$C_{\text{sp}}(T^{\text{card}})$	204	204	320	320	640
$C_{\text{id}}(T^{\text{card}})$	40	48	56	64	72

$I := \{1, \dots, n\}$ differ in the edge concentrated case. Construction 4.3 yields the block $\mathscr{H}$-trees T^{card} or T^{geo}, respectively, whose leaves partition the product index set $I \times I$. The parameter η in the admissibility condition is $\eta := 1.0$ and the minimal blocksize is $n_{\min} := 32$. For $n = 3058$ degrees of freedom the partitioning (geometrically and cardinality balanced) is depicted in Fig. 5. We should mention that the left picture in Fig. 5 is slightly misleading because the structure of the partitioning is not as regular as it seems: blocks $r \times s$ with $\#r \le n_{\min}$ or $\#s \le n_{\min}$ and $\#r \gg n_{\min}$ or $\#s \gg n_{\min}$ are not visible but they appear frequently. The sparsity C_{sp} and the idempotency C_{id} of the trees T^{card} and T^{geo} are given in Table 9. Again we observe that the sparsity and idempotency is bounded for the geometrically balanced tree T^{geo} while the sparsity of the cardinality balanced tree T^{card} seems to be $C_{\text{sp}}(T^{\text{card}}) = \mathcal{O}(\sqrt{n})$. In Fig. 5 we find that the maximal sparsity appears only in a few rows or columns of the matrix and indeed the numerical results in Table 11 indicate that the (formatted) inversion is of complexity $\mathcal{O}(n \log(n)^2)$. Since the

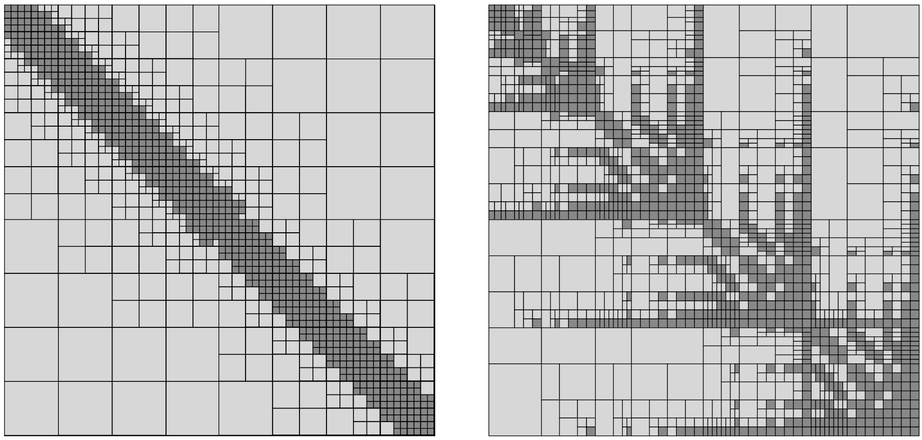


Figure 5. The partitioning of the product index set $I \times I$ in the edge concentrated case for $n = 3058$ degrees of freedom; to the left the geometrically balanced and to the right cardinality balanced case. $R(k)$-blocks are light grey and full matrix blocks are dark grey

Table 10. Left: time (in seconds) for the (formatted) inversion on the boundary concentrated grid. Right: relative error $\|I - A\ \mathrm{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on the edge concentrated grid

$n =$	6129	12272	24559	49134	98285
k = 1	15.01	36.01	8.7 + 1	2.1 + 2	4.8 + 2
k = 2	15.76	38.74	9.4 + 1	2.3 + 2	5.4 + 2
k = 3	16.83	41.79	1.0 + 2	2.5 + 2	6.1 + 2
k = 4	17.93	44.80	1.1 + 2	2.7 + 2	6.4 + 2
k = 5	18.94	47.34	1.2 + 2	2.9 + 2	6.9 + 2
k = 6	19.56	49.50	1.2 + 2	3.1 + 2	7.4 + 2
k = 7	19.78	50.55	1.3 + 2	3.2 + 2	7.7 + 2

$n =$	6129	12272	24559	49134	98285
k = 1	3.2−2	3.9−2	4.4−2	4.7−2	4.9−2
k = 2	4.1−3	4.3−3	4.6−3	4.7−3	4.9−3
k = 3	4.3−5	4.6−5	4.7−5	4.9−5	4.9−5
k = 4	5.3−6	6.2−6	6.4−6	6.8−6	6.9−6
k = 5	1.1−8	1.3−8	1.3−8	1.3−8	1.3−8
k = 6	5.0−11	5.8−11	5.8−11	5.9−11	6.2−11
k = 7	1.9−14	2.8−14	3.5−14	4.5−14	5.2−14

Table 11. Left: time (in seconds) for the (formatted) inversion on the edge concentrated grid for the cardinality balanced tree T^{card}. Right: relative error $\|I - A\ \mathrm{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on the edge concentrated grid

$n =$	6129	12272	24559	49134	98285
k = 1	42.21	1.1 + 2	2.5 + 2	5.5 + 2	1.3 + 3
k = 2	50.69	1.2 + 2	3.0 + 2	6.6 + 2	1.6 + 3
k = 3	57.24	1.5 + 2	3.6 + 2	8.2 + 2	2.1 + 3
k = 4	67.15	1.7 + 2	4.3 + 2	9.8 + 2	2.5 + 3

$n =$	6129	12272	24559	49134	98285
k = 1	4.6−2	5.5−2	5.5−2	6.5−2	7.0−2
k = 2	5.5−3	6.3−3	6.4−3	6.7−3	6.9−3
k = 3	1.3−3	1.5−3	1.6−3	1.7−3	1.7−3
k = 4	2.9−5	3.2−5	3.5−5	3.7−5	3.7−5

(formatted) inversion with the cardinality balanced tree is by a factor of $2 - 3$ slower than with the geometrically balanced tree, it is advisable to use the latter one for which we have proven the desired estimates of the complexity.

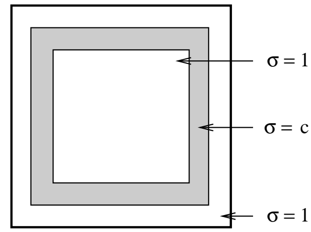
5.5 A Differential Operator with "Jumping Coefficients"

In this section we replace the Laplacian $-\Delta = -\mathrm{div}\nabla$ by the operator $-\Delta_c$,

$$-\Delta_c[u](x) := -(\text{div } \sigma(c,x)\nabla)[u](x),$$

where the function $\sigma(c,x) : \mathbb{R} \times [0,1]^2 \to \mathbb{R}$ is defined by

$$\sigma(c,x) := \begin{cases} c & \text{if } x \in [0.1, 0.9]^2 \setminus [0.2, 0.8]^2, \\ 1 & \text{otherwise.} \end{cases}$$



The construction of the $\mathscr{H}$-tree T_I and the block $\mathscr{H}$-tree T is the same as in Section 5.2. Consequently, the sparsity and idempotency constants are the same. Moreover, the complexity for the formatted inversion is the same in the sense that the numbers coincide exactly with those of Tables 3–5. In Table 12 we present the approximation error $\|I - A\,\text{Inv}(A)\|$ in the spectral norm for the formatted inverse $\text{Inv}(A)$. In this first example the coefficient $\sigma(c,x)$ is chosen in a structured way as it may occur, e.g., for technical devices. As a second example we choose the coefficient $\sigma(c,x)$ in a stochastic way: for each element τ in our triangulation we define a random real number $c_\tau \in [1,c]$ and let

$$\sigma(c,x) := c_\tau, \qquad \text{for } x \in \tau,$$

i.e., $\sigma(c,x)$ is piecewise constant. Table 13 presents the approximation error $\|I - A\,\text{Inv}(A)\|$ in the spectral norm for the formatted inverse $\text{Inv}(A)$. The approximation error is (roughly) the same as for the Laplace operator (cf. Table 3).

Table 12. The relative error $\|I - A\,\text{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on the uniform grid where the coefficient is $c = 10$ left and $c = 100$ right

$c = 10$	$n = 4096$	$n = 16384$	$n = 65536$
$k = 1$	6.7	26.3	57.3
$k = 5$	4.2−3	3.6−2	1.6−1
$k = 9$	3.2−5	2.5−4	9.7−4
$k = 13$	7.0−7	3.4−6	1.8−5
$k = 17$	6.1−11	2.5−9	1.4−8

$c = 100$	$n = 4096$	$n = 16384$	$n = 65536$
$k = 1$	111	199.9	179.7
$k = 5$	3.1−2	1.79	3.19
$k = 9$	3.9−4	4.2−3	1.7−2
$k = 13$	1.8−6	1.4−4	1.8−4
$k = 17$	1.1−10	2.4−8	4.3−7

Table 13. The relative error $\|I - A\,\text{Inv}(A)\|$ in the spectral norm for the (formatted) inverse on the uniform grid where the bound for the random coefficient is $c = 10$ left and $c = 100$ right

$c = 10$	$n = 4096$	$n = 16384$	$n = 65536$
$k = 1$	2.77	10.15	31.04
$k = 5$	2.1−3	9.9−3	4.9−2
$k = 9$	9.7−6	6.5−5	3.1−4
$k = 13$	3.8−7	1.3−6	4.9−6
$k = 17$	9.6−11	3.4−9	1.4−8

$c = 100$	$n = 4096$	$n = 16384$	$n = 65536$
$k = 1$	3.00	10.59	32.96
$k = 5$	2.4−3	1.1−2	5.1−2
$k = 9$	1.6−5	6.9−5	3.2−4
$k = 13$	1.5−7	1.1−6	4.0−6
$k = 17$	7.8−11	4.8−9	9.5−9

References

[1] Bebendorf, M., Hackbusch, W.: Existence of $\mathscr{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. Preprint No. 21, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2002. To appear in Numer. Math.

[2] Börm, S., Grasedyck, L.: Low-rank approximation of integral operators by interpolation. Preprint No. 72, Leipzig: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2002.

[3] Börm, S., Grasedyck, L., Hackbusch, W.: Introduction to hierarchical matrices with applications. EABE *27*, 403–564 (2003).

[4] Börm, S., Löhndorf, M., Melenk, J.: Approximation of integral operators by variable-order interpolation. Preprint No. 82, Leipzig: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2002.

[5] Golub, G. H., Van Loan, C. F.: Matrix computations. Baltimore, London: Johns Hopkins University Press, 1996.

[6] Graham, I. G., Hackbusch, W., Sauter, S. A.: Finite elements on degenerated meshes: inverse-type inequalities and applications. Preprint No. 102, Leipzig: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2002.

[7] Grasedyck, L.: Theorie und Anwendungen Hierarchischer Matrizen. Doctoral thesis, Universität zu Kiel, Germany, 2001.

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

[9] Hackbusch, W.: Iterative solution of large sparse systems. New York: Springer-Verlag, 1994.

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

[11] Hackbusch, W., Khoromskij, B. N., Sauter, S. A.: On $\mathscr{H}^2$-matrices. In: Bungartz, H.-J., Hoppe, R.H.W., Zenger, C. (eds.) Lectures on applied mathematics. Berlin: Springer-Verlag, 2000.

[12] Le Borne, S.: $\mathscr{H}$-matrices for convection-diffusion problems with constant convection. Computing *70*, 261–274 (2003)

L. Grasedyck
W. Hackbusch
Max-Planck-Institute for Mathematics in the Sciences
Inselstr. 22-26
D-04103 Leipzig Germany
e-mail: {lgr,wh}@mis.mpg.de