Kompaktkurs

Lineare Gleichungssysteme Hierarchische Matrizen

M. Bebendorf, O. Steinbach

SIMNET Styria

Das Land Steiermark

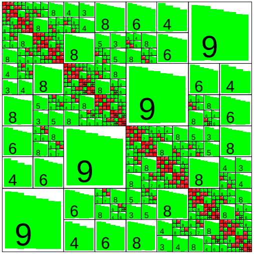
→ Wissenschaft und Forschung

Teil 2: Hierarchische Matrizen

Mario Bebendorf, Universität Leipzig
bebendorf@math.uni-leipzig.de

Fast Summation Methods

- *Fast multipole methods*: Rokhlin 1985;
- *Panel clustering*: Hackbusch/Nowak 1989;
- *Mosaic-skeleton methods*: Tyrtyshnikov 1996;
- *Hierarchical matrices*: Hackbusch 2000 and Hackbusch/Khoromskij 2001.



Main idea: Approximation errors can be tolerated as long their size is of the order of discretization error.
Aim: Solution of linear systems with complexity $\mathcal{O}(n \log^* n)$.
Here: Emphasis is laid on algorithmic aspects; approximation theory is neglected.

Low-Rank Matrices

Let $m, n \in \mathbb{N}$ and $A \in \mathbb{C}^{m\times n}$ be a matrix. The rank of A is the dimension of its range

$$\operatorname{rank} A := \dim \{Ax \in \mathbb{C}^m, \, x \in \mathbb{C}^n\}.$$

Theorem

Let $m, n, k \in \mathbb{N}$. Then it holds that

(a) $\operatorname{rank} A \le \min\{m, n\}$ *for all* $A \in \mathbb{C}^{m\times n}$

(b) $\operatorname{rank}(AB) \le \min\{\operatorname{rank} A, \operatorname{rank} B\}$ *for all* $A \in \mathbb{C}^{m\times p}$ *and all* $B \in \mathbb{C}^{p\times n}$

(c) $\operatorname{rank}(A+B) \le \operatorname{rank} A + \operatorname{rank} B$ *for all* $A, B \in \mathbb{C}^{m\times n}$

We denote the set of matrices $A \in \mathbb{C}^{m\times n}$ having at most k linearly independent rows or columns by

$$\mathbb{C}_k^{m\times n} := \left\{A \in \mathbb{C}^{m\times n} : \operatorname{rank} A \le k\right\}.$$

NOTE:

- $\mathbb{C}_k^{m\times n}$ is not a linear space. The rank of the sum of two rank-k matrices is in general only bounded by $2k$.
- The rank of each sub-block of $A \in \mathbb{C}_k^{m\times n}$ is bounded by k.

Efficient representation

Since among the n columns of $A \in \mathbb{C}_k^{m \times n}$ only k are sufficient to represent the whole matrix by linear combination, the entrywise representation of A contains redundancies which can be removed by changing to another kind of representation.

Theorem

A matrix $A \in \mathbb{C}^{m \times n}$ belongs to $\mathbb{C}_k^{m \times n}$ if and only if there are matrices $U \in \mathbb{C}^{m \times k}$ and $V \in \mathbb{C}^{n \times k}$ such that

$$A = UV^H. \tag{1}$$

The representation (1) of matrices from $\mathbb{C}_k^{m \times n}$ is called **outer-product form**. If u_i, v_i, $i = 1, \dots, k$, denote the columns of U and V, respectively, then (1) can be equivalently written as

$$A = \sum_{i=1}^{k} u_i v_i^H.$$

Hence, instead of storing the $m \cdot n$ entries of $A \in \mathbb{C}_k^{m \times n}$, we can equally store the vectors u_i, v_i, $i = 1, \dots, k$, which require $k(m + n)$ units of storage.

In addition to reducing the storage requirements, the outer-product form (1) also facilitates matrix-vector multiplications:

$$Ax = UV^H x = U(V^H x),$$

i.e, instead of computing the update $y := y + Ax$ in the usual (entrywise) way, A can alternatively be multiplied by x using the following two-step procedure:

(a) define $z := V^H x \in \mathbb{C}^k$

(b) compute $y := y + Uz$.

Hence, instead of $2m \cdot n$ arithmetic operations which are required in the entrywise representation, the outer-product form amounts to $2k(m+n) - k$ operations.
Assume for a moment that $m = n$.

- Outer-product form replaces one dimension in the complexity of matrices from $\mathbb{C}_k^{m \times n}$ by $2k$, i.e., instead of $m \cdot n = m^2$ we have $k(m+n) = 2km$
- this representation is not advantageous for large k.

By the following definition we characterize matrices for which the outer-product form is advantageous compared with the entrywise representation.

Definition

A matrix $A \in \mathbb{C}_k^{m\times n}$ *is called a* **matrix of low rank** *if*

$$k(m+n) < m\cdot n.$$

Besides arithmetic operations also the norm of a matrix is often required. The **Frobenius norm**

$$\|A\|_F := \sqrt{\operatorname{trace} A^H A} = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2\right)^{1/2}$$

of $A \in \mathbb{C}_k^{m\times n}$ can be computed with $2k^2(m+n)$ operations since

$$\|UV^H\|_F^2 = \sum_{i,j=1}^k (u_i^H u_j)(v_i^H v_j). \tag{2}$$

Similarly, the **spectral norm**

$$\|A\|_2 := \sqrt{\rho(A^H A)} = \sqrt{\rho(VU^H UV^H)} = \sqrt{\rho(U^H UV^H V)},$$

where $\rho(A)$ denotes the spectral radius of A, can be evaluated with $\mathcal{O}(k^2(m+n))$ arithmetic operations.

Sometimes it is useful that U and V have orthonormal columns; i.e., it holds that $U^H U = I_k = V^H V$. In this case we have to introduce an additional diagonal coefficient matrix $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_k) \in \mathbb{R}^{k \times k}$ and replace (1) by the **singular value decomposition**

$$A = U\Sigma V^H = \sum_{i=1}^{k} \sigma_i u_i v_i^H. \tag{3}$$

If the representation (3) with matrices U, V having orthonormal columns is employed, the computation of the Frobenius norm simplifies to

$$\|U\Sigma V^H\|_F = \|\Sigma\|_F = \sqrt{\sum_{i=1}^{k} \sigma_i^2},$$

which requires only $\mathcal{O}(k)$ operations. Since the spectral norm is unitarily invariant, too, we have

$$\|U\Sigma V^H\|_2 = \|\Sigma\|_2 = \sigma_1,$$

where we assume that the singular values σ_i, $i = 1, \dots, k$, of A are non-increasingly ordered; i.e., it holds that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k$.

Adding and multiplying low-rank matrices

We consider the multiplication of low-rank matrices $A \in \mathbb{C}^{m\times p}_{k_A}$ and $B \in \mathbb{C}^{p\times n}_{k_B}$ in outer-product representation

$$A = U_A V_A^H \quad \text{and} \quad B = U_B V_B^H.$$

The rank of the product AB is bounded by $\min\{k_A, k_B\}$. Hence, the outer-product form will be advantageous for the product AB as well. There are two possibilities for computing $AB = UV^H$:

(a) $U := U_A(V_A^H U_B)$ and $V := V_B$ using $2k_Ak_B(m+p) - k_B(m+k_A)$ operations

(b) $U := U_A$ and $V := V_B(U_B^H V_A)$ using $2k_Ak_B(p+n) - k_A(n+k_B)$ operations.

Depending on the quantities k_A, k_B, m, and n, either representation should be chosen.

If exactly one of the matrices A or B is stored entrywise, say B, we have the following outer-product representation of $AB = UV^H$, where $U := U_A \in \mathbb{C}^{m\times k_A}$ and $V := B^H V_A \in \mathbb{C}^{n\times k_A}$. This requires $k_A(2p-1)n$ operations.

If A and B are to be added, the sum $A + B$ will have the following outer-product representation

$$A + B = UV^H$$

with $U := [U_A, U_B] \in \mathbb{C}^{m \times k}$ and $V := [V_A, V_B] \in \mathbb{C}^{n \times k}$, which guarantees that

$$\operatorname{rank}(A + B) \leq k_A + k_B =: k. \tag{4}$$

Hence, apart from the reorganization of the data structure, no numerical operations are required for adding two matrices in outer-product form.
A lower bound than (4) for the rank of $A + B$ cannot be found for general low-rank matrices. Hence, the rank of $A + B$ will be considerably larger than the ranks of A and B although $A + B$ might be close to a matrix of a much smaller rank.

Approximation by low-rank matrices

Although matrices usually have full rank, they can often be approximated by matrices having a much lower rank. The following theorem states that the closest matrix in $\mathbb{C}_k^{m\times n}$ to a given matrix from $\mathbb{C}^{m\times n}$, $m \geq n$, can be obtained from the singular value decomposition (SVD) $A = U\Sigma V^H$ with $U^H U = I_n = V^H V$ and a diagonal matrix $\Sigma \in \mathbb{R}^{n\times n}$ with entries $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$. Interestingly, this result is valid for any unitarily invariant norm.

Theorem

Let the SVD $A = U\Sigma V^H$ of $A \in \mathbb{C}^{m\times n}$, $m \geq n$, be given. Then for $k \in \mathbb{N}$ satisfying $k \leq n$ it holds that

$$\min_{M\in\mathbb{C}_k^{m\times n}} \|A - M\| = \|A - A_k\| = \|\Sigma - \Sigma_k\|, \tag{5}$$

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

Note that the approximant $U\Sigma_k V^H$ has the representation (3). If the outer-product representation is preferred, either U or V has to be multiplied by Σ_k.

If the spectral norm $\|\cdot\|_2$ is used in the previous theorem, then

$$\|A - A_k\|_2 = \sigma_{k+1}.$$

In the case of the Frobenius norm one has instead

$$\|A - A_k\|_F^2 = \sum_{\ell=k+1}^{n} \sigma_\ell^2.$$

The information about the error $\|A - A_k\| = \|\Sigma - \Sigma_k\|$ can also be used in the opposite way. If on the other hand a relative accuracy $\varepsilon > 0$ of the approximant A_k is prescribed, i.e.,

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

then due to (5) the required rank $k(\varepsilon)$ is given by

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

In order to find the optimum approximant A_k, it remains to compute the singular value decomposition, which requires $\mathcal{O}(mn^2)$ operations for general matrices $A \in \mathbb{C}^{m\times n}$, $m \geq n$. If the given matrix A has low rank, then its SVD can be computed with significantly less operations.

Singular value decomposition of low-rank matrices

For matrices $A = UV^H \in \mathbb{C}_k^{m\times n}$ it is possible to compute an SVD with complexity $\mathcal{O}(k^2(m+n))$.

Assume we have computed the QR decompositions

$$U = Q_U R_U \quad \text{and} \quad V = Q_V R_V$$

of $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$, respectively. Note that this can be done with $4k^2(m+n)$ operations. The outer-product of the two $k \times k$ upper triangular matrices R_U and R_V is then decomposed using the SVD of $R_U R_V^H$:

$$R_U R_V^H = \hat{U}\hat{\Sigma}\hat{V}^H.$$

Computing $R_U R_V^H$ needs $k^2(k+1)$ operations and the cost of the SVD amount to $21k^3$ operations. Since $Q_U\hat{U}$ and $Q_V\hat{V}$ both are unitary,

$$A = UV^H = (Q_U\hat{U})\hat{\Sigma}(Q_V\hat{V})^H$$

is an SVD of A. The number of arithmetic operations of the SVD of a rank-k matrix sum up to

QR decomposition of U and V	$4k^2(m+n)$
Computing $R_U R_V^H$	$k^2(k+1)$
SVD of $R_U R_V^H$	$21k^3$
Computing $Q_U\hat{U}$ and $Q_V\hat{V}$	$k(2k-1)(m+n)$
	$\sim\ 6k^2(m+n) + 22k^3$

operations.

Approximate addition of low-rank matrices

When computing the sum of two low-rank matrices, we have to deal with the problem that $\mathbb{C}_k^{m\times n}$ is not a linear space. The sum $A+B$ of two matrices from $\mathbb{C}_k^{m\times n}$ might, however, be close to a matrix of a much smaller rank. In this case the sum of two low-rank matrices can be truncated to rank k using the SVD of low-rank matrices from the last section. This truncation will be referred to as the **rounded addition**.

Theorem

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

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

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

In some applications it may also occur that the sum of several low-rank matrices $A_i\in\mathbb{C}_{k_i}^{m\times n}$, $i=1,\dots,\ell$, has to be rounded. In this case, the complexity analysis reveals a factor $(\sum_{i=1}^{\ell} k_i)^2$ in front of $m+n$. In order to avoid this, we gradually compute the rounded sum pairwise.

Exploiting the SVD for the rounded addition

The rounded addition is the most time-consuming part in the arithmetic of hierarchical matrices. The numerical effort can be reduced if the SVD representation is used for instance. Assume that

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

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

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

Assume that $k_A \geq k_B$. In order to reestablish a representation of type (3), we have to orthogonalize the columns of the matrices $[U_A, U_B]$ and $[V_A, V_B]$. Let $X_U := U_A^H U_B \in \mathbb{C}^{k_A \times k_B}$ and $Y_U := U_B - U_A X_U \in \mathbb{C}^{m \times k_B}$. Furthermore, let $Q_U R_U = Y_U$, $Q_U \in \mathbb{C}^{m \times k_B}$, be a QR decomposition of Y_U. Then

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

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

Similarly,

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

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

$$A + B = [U_A, Q_U] \begin{bmatrix} \Sigma_A + X_U \Sigma_B X_V^H & X_U \Sigma_B R_V^H \\ R_U \Sigma_B X_V^H & R_U \Sigma_B R_V^H \end{bmatrix} [V_A, Q_V]^H.$$

From this point on one proceeds in the same way as for the SVD of low-rank matrices. For the complexity analysis we concentrate on those terms which depend on m. The orthogonalization of $[U_A, U_B]$ using the previous method requires

Computing X_U	$2k_A k_B m$
Computing Y_U	$(k_A + 1)k_B m$
Decomposing Y_U	$4k_B^2 m$
	$(3k_A + 4k_B + 1)k_B m$

operations while the orthogonalization of $[U_A, U_B]$ using the QR decomposition needs $4(k_A + k_B)^2 m$ operations.

If $k := k_A = k_B$, then the proposed variant requires $(7k+1)km$ operations, while $16k^2m$ operations are needed to decompose $[U_A, U_B]$.

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

The presented CPU times are the times for 10 000 additions with accuracy $\varepsilon = 1_{10}{-}2$.

Agglomerating low-rank blocks

We will come across the problem of unifying neighboring blocks to a single one for the purpose of saving memory. This operation will be referred to as **agglomeration**. Assume a 2 × 2 block matrix

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \approx UV^H$$

consisting of four low-rank matrices $A_i = U_i V_i^H$, $i = 1, \dots, 4$, each having rank at most k is to be approximated by a single matrix $A = UV^H \in \mathbb{C}_k^{m\times n}$. Since

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

this problem may be regarded as a rounded addition of four low-rank matrices. Therefore, a best approximation in $\mathbb{C}_k^{m\times n}$ can be computed using the SVD of low-rank matrices.

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

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

where

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

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

Computing the QR decompositions	$2 \cdot 7k^2(m+n)$
Computing $R_{\hat{U}} R_{\hat{V}}^H$	$7k(2k)^2$
Computing the SVD of $R_{\hat{U}} R_{\hat{V}}^H$	$21(4k)^3$
Building the unitary factors	$4k^2(m+n)$
	$\sim\ 18k^2(m+n) + 1372k^3$

The amount of operations can be reduced if each of the matrices $[A_1, A_2]$ and $[A_3, A_4]$ is agglomerated before agglomerating the results.

Degenerate Kernels

Typically, only sub-blocks of $A \in \mathbb{C}^{I \times J}$, $I = \{1, \dots, m\}$ and $J := \{1, \dots, n\}$, can be approximated by low-rank matrices.

- COMMON: the ith component of a vector $x \in \mathbb{C}^I$ is denoted by x_i
- GENERALIZATION: If $t \subset I$, then $x_t \in \mathbb{C}^t$ denotes the restriction of x to the indices in t. Consequently, A_{ts} or A_b denotes the restriction of a given matrix $A \in \mathbb{C}^{m \times n}$ to the indices in $b := t \times s$, where $t \subset I$ and $s \subset J$.

In this part we consider matrices $A \in \mathbb{R}^{I \times J}$ with blocks A_{ts} of the form

$$A_{ts} = \Lambda_{1,t} \mathcal{A} \Lambda_{2,s}^*,$$

which arise from the discretization of integral operators

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

Here, $\Omega \subset \mathbb{R}^d$ denotes the domain of integration and μ is an associated measure. If Ω is a $(d-1)$-dimensional manifold in $\mathbb{R}^d$ for instance, then μ denotes the surface measure. Furthermore,

$$(\Lambda_{1,t} f)_i = \int_\Omega f(x) \psi_i(x) \, \mathrm{d}\mu_x, \quad (\Lambda_{2,s} f)_j = \int_\Omega f(x) \varphi_j(x) \, \mathrm{d}\mu_x$$

and $\Lambda_{2,s}^* : \mathbb{R}^s \to L^2(\Gamma)$ is the adjoint of $\Lambda_{2,s} : L^2(\Gamma) \to \mathbb{R}^s$ defined by

$$(\Lambda_{2,s}^* z, f)_{L^2(\Gamma)} = z^T (\Lambda_{2,s} f) \quad \text{for all } z \in \mathbb{R}^s,\ f \in L^2(\Gamma).$$

Each set of rows $t \subset I$ and each set of columns $s \subset J$ is connected with subdomains

$$Y_t := \bigcup_{i \in t} Y_i \quad \text{and} \quad X_s = \bigcup_{j \in s} X_j$$

which are the union of supports $Y_i \subset \Omega$ and $X_j \subset \Omega$ of finite element basis functions ψ_i, φ_j defined on the computational domain Ω.

Assume that the kernel function κ is degenerate on $Y_t \times X_s$. Later on, we will find conditions on t and s for this assumption to hold.

Definition

Let $D_1, D_2 \subset \mathbb{R}^d$ be two domains. A kernel function $\kappa : D_1 \times D_2 \to \mathbb{R}$ is called **degenerate** *if $k \in \mathbb{N}$ and functions $u_\ell : D_1 \to \mathbb{R}$ and $v_\ell : D_2 \to \mathbb{R}$, $\ell = 1, \dots, k$, exist such that*

$$\kappa(x, y) = \sum_{\ell=1}^{k} u_\ell(x) v_\ell(y), \quad x \in D_1,\ y \in D_2.$$

The number k is called **degree of degeneracy***.*

The rank of A_{ts} is bounded by k. To see this, let

$$a_\ell = \Lambda_{1,t} v_\ell \in \mathbb{R}^t \quad \text{and} \quad b_\ell = \Lambda_{2,s} u_\ell \in \mathbb{R}^s, \quad \ell = 1, \dots, k.$$

For $z \in \mathbb{R}^s$ we have

$$b_\ell^T z = (u_\ell, \Lambda_{2,s}^* z)_{L^2(X_s)}.$$

Since for $y \in Y_t$

$$\begin{aligned}(\mathcal{A}\Lambda_{2,s}^* z)(y) &= \int_{X_s} \kappa(x,y)(\Lambda_{2,s}^* z)(x)\,\mathrm{d}\mu_x = \int_{X_s} \sum_{\ell=1}^k u_\ell(x) v_\ell(y)(\Lambda_{2,s}^* z)(x)\,\mathrm{d}\mu_x \\ &= \sum_{\ell=1}^k v_\ell(y) \int_{X_s} u_\ell(x)(\Lambda_{2,s}^* z)(x)\,\mathrm{d}\mu_x = \sum_{\ell=1}^k v_\ell(y) b_\ell^T z,\end{aligned}$$

we obtain for the sub-block A_{ts} of A

$$A_{ts} = \Lambda_{1,t}\mathcal{A}\Lambda_{2,s}^* = \Lambda_{1,t} \sum_{\ell=1}^k v_\ell b_\ell^T = \sum_{\ell=1} (\Lambda_{1,t} v_\ell) b_\ell^T = \sum_{\ell=1}^k a_\ell b_\ell^T.$$

Therefore, degenerate kernels lead to low-rank matrices if t and s are large enough compared with k; i.e., if $k(|t| + |s|) < |t||s|$.

Asymptotically smooth kernels

If $\mathcal{A}$ is an elliptic operator, then its kernel function κ is asymptotically smooth.

Definition

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

$$|\partial_y^\alpha \kappa(x, y)| \le c p! \gamma^p r^{-p} \sup_{z \in B_r(y)} |\kappa(x, z)| \quad \textit{for all } y \in \mathbb{R}^d \setminus \{x\},$$

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

Let $\kappa : D_1 \times D_2 \to \mathbb{R}$ be analytic with respect to its second argument y and let ξ_{D_2} denote the Chebyshev center of D_2. Then κ has a Taylor expansion

$$\kappa(x, y) = \sum_{|\alpha| < p} \frac{1}{\alpha!} \partial_y^\alpha \kappa(x, \xi_{D_2})(y - \xi_{D_2})^\alpha + R_p(x, y),$$

where

$$R_p(x, y) := \sum_{|\alpha| \ge p} \frac{1}{\alpha!} \partial_y^\alpha \kappa(x, \xi_{D_2})(y - \xi_{D_2})^\alpha$$

denotes the remainder of the expansion, which converges to zero for $p \to \infty$. The rate of convergence can however be arbitrarily bad.

Note that

$$T_p[\kappa](x,y) := \sum_{|\alpha|<p} \frac{1}{\alpha!} \partial_y^\alpha \kappa(x, \xi_{D_2})(y-\xi_{D_2})^\alpha$$

is a degenerate kernel approximation. Since $T_p[\kappa](x,\cdot) \in \Pi_{p-1}^d$, the degree of degeneracy is the dimension of the space of d-variate polynomials of order at most $p-1$

$$k = \dim \Pi_{p-1}^d \le p^d.$$

The importance of asymptotic smoothness is that it leads to exponential convergence of the Taylor series if D_1 and D_2 are far enough away from each other. For the following lemma we assume that

$$\eta \operatorname{dist}(\xi_{D_2}, D_1) \ge \rho_{D_2}, \tag{6}$$

where $\eta > 0$ and ξ_{D_2} is the Chebyshev center of $D_2 \subset \mathbb{R}^d$, i.e., the center of the ball with minimum radius ρ_{D_2} containing D_2.

Lemma

Assume that (6) *holds with* $\eta > 0$ *satisfying* $2\gamma\sqrt{d}\eta < 1$. *If* κ *is asymptotically smooth on the convex set* D_2 *with respect to* y, *then with* $r := |x - \xi_{D_2}|/2$ *for all* $x \in D_1$ *and* $y \in D_2$ *it holds that*

$$|\kappa(x,y) - T_p[\kappa](x,y)| \le \frac{(2\gamma\sqrt{d}\eta)^p}{1-2\gamma\sqrt{d}\eta} \sup_{z \in B_r(\xi_{D_2})} |\kappa(x,z)|.$$

Proof. For the remainder R_p it holds that

$$
\begin{aligned}
|R_p(x,y)| &\le \sum_{|\alpha|\ge p} \frac{1}{\alpha!} |\partial_y^\alpha \kappa(x,\xi_{D_2})| |(y-\xi_{D_2})^\alpha| \\
&\le c \sup_{z\in B_r(\xi_{D_2})} |\kappa(x,z)| \sum_{|\alpha|\ge p} \frac{(2\gamma)^{|\alpha|}|\alpha|!}{\alpha! |x-\xi_{D_2}|^{|\alpha|}} |(y-\xi_{D_2})^\alpha| \\
&= c \sup_{z\in B_r(\xi_{D_2})} |\kappa(x,z)| \sum_{\ell=p}^{\infty} \left(\frac{2\gamma}{|x-\xi_{D_2}|}\right)^\ell \sum_{|\alpha|=\ell} \binom{\ell}{\alpha} |(y-\xi_{D_2})^\alpha| \\
&\le c \sup_{z\in B_r(\xi_{D_2})} |\kappa(x,z)| \sum_{\ell=p}^{\infty} \left(2\gamma\sqrt{d}\frac{|y-\xi_{D_2}|}{|x-\xi_{D_2}|}\right)^\ell \\
&\le c \sup_{z\in B_r(\xi_{D_2})} |\kappa(x,z)| \sum_{\ell=p}^{\infty} (2\gamma\sqrt{d}\eta)^\ell \\
&\le c \frac{(2\gamma\sqrt{d}\eta)^p}{1-2\gamma\sqrt{d}\eta} \sup_{z\in B_r(\xi_{D_2})} |\kappa(x,z)|
\end{aligned}
$$

due to $\sum_{|\alpha|=\ell} \binom{\ell}{\alpha} |\xi^\alpha| = (\sum_{i=1}^d |\xi_i|)^\ell \le d^{\ell/2}|\xi|^\ell$ for all $\xi \in \mathbb{R}^d$. ■

The previous lemma shows that the Taylor expansion of asymptotically smooth kernels converges exponentially with convergence rate $2\gamma\sqrt{d}\eta < 1$. Thus, $p \sim |\log\varepsilon|$ is required to achieve a given approximation accuracy $\varepsilon > 0$. For the degree k of degeneracy of $T_p[\kappa]$ it follows that

$$k \sim p^d \sim |\log\varepsilon|^d.$$

Note that if κ is asymptotically smooth only with respect to the first argument x, then ρ_{D_2} has to be replaced by ρ_{D_1} in (6). If κ is asymptotically smooth with respect to both variables, then the symmetric condition

$$\min\{\rho_{D_1}, \rho_{D_2}\} \leq \eta \operatorname{dist}(D_1, D_2) \tag{7}$$

is sufficient.

In order to be able to approximate the block A_{ts}, we have to satisfy the last condition for $D_1 = Y_t$ and $D_2 = X_s$. A block $t \times s$ will therefore be called **admissible** if the condition

$$\min\{\operatorname{diam} Y_t, \operatorname{diam} X_s\} \leq \eta \operatorname{dist}(Y_t, X_s) \tag{8}$$

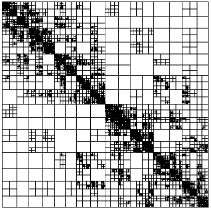
is satisfied.

Matrix partitioning

When constructing partitions, we have to account for two contrary aims.

- partition has to be fine enough such that most of the blocks can be successfully approximated;
- the number of blocks must be as small as possible in order to be able to guarantee efficient arithmetic operations.

Finding an optimal partition is a difficult task since the set of all possible partitions is too large to be searched for. Instead of searching for the best partition, we will therefore construct partitions which are quasi-optimal in the sense that they can be computed with almost linear costs and allow approximants of logarithmic-linear complexity.



Note that at least for the diagonal entries (i, i), $i \in I$, both conditions (8) will always be violated. In order to guarantee that A_b, $(i, i) \in b$, has low rank, the dimensions of A_b therefore have to be small.

This leads to the following definition.

Definition

A partition P is called **admissible** *if each block $t \times s \in P$ is either admissible or small; i.e., the cardinalities $|t|$ and $|s|$ of t and s satisfy* $\min\{|t|, |s|\} \leq n_{\min}$ *with a given minimal dimension $n_{\min} \in \mathbb{N}$.*

Tensor vs. hierarchical partitions

In this section we assume that $I = J$. For the comparison of tensor and hierarchical partitions we will investigate the memory consumption and the number of arithmetic operations required to multiply such matrices by a vector if

(a) each diagonal block is stored as a dense matrix;

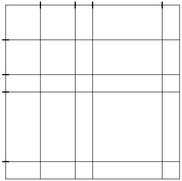
(b) all other blocks $t \times s$ are assumed to be admissible and are stored as rank-1 matrices.

Let us first consider the case of tensor partitions

$$P = P_I \times P_I = \{t_k \times t_\ell : t_k, t_\ell \in P_I\},$$

where $P_I := \{t_k,\ k = 1, \ldots, p\}$ is a partition of the index set I; i.e.,

$$I = \bigcup_{k=1}^{p} t_k \quad \text{and} \quad t_k \cap t_\ell = \varnothing \quad \text{for } k \neq \ell.$$



Storing A requires $\sum_{k=1}^{p} |t_k|^2$ units of storage for the diagonal and

$$\sum_{k\neq\ell} |t_k| + |t_\ell| = 2 \sum_{k=1}^{p} \sum_{\substack{\ell=1\\ \ell\neq k}}^{p} |t_k| = 2(p-1) \sum_{k=1}^{p} |t_k| = 2(p-1)n$$

units for the off-diagonal blocks. Due to the Cauchy-Schwarz inequality

$$n^2 = \left(\sum_{k=1}^{p} |t_k| \right)^2 \leq p \sum_{k=1}^{p} |t_k|^2,$$

at least $n^2/p + 2(p-1)n$ units of storage are necessary to hold A. The minimum of the last expression is attained for $p = \sqrt{n/2}$ resulting in a minimum amount of storage of order $n^{3/2}$. Hence, the required amount of storage resulting from tensor product partitions is not competitive.

Let us now check whether a hierarchical partition leads to almost linear complexity. We restrict ourselves to the case $n = 2^p$ for some $p \in \mathbb{N}$.

Assume that t has already been generated from I after ℓ subdivisions. Subdividing $t = \{i_1, \dots, i_{2^{p-\ell}}\}$ into two parts

$$t_1 = \{i_1, \dots, i_{2^{p-\ell-1}}\} \quad \text{and} \quad t_2 = \{i_{2^{p-\ell-1}+1}, \dots, i_{2^{p-\ell}}\}$$

of equal size, we obtain a 2×2 block partition of $A_{tt} \in \mathbb{R}^{t \times t}$:

$$A_{tt} = \begin{bmatrix} A_{t_1 t_1} & A_{t_1 t_2} \\ A_{t_2 t_1} & A_{t_2 t_2} \end{bmatrix}, \tag{9}$$

where $A_{t_i t_j} \in \mathbb{R}^{t_i \times t_j}$, $i, j = 1, 2$. The diagonal blocks $A_{t_1 t_1}$ and $A_{t_2 t_2}$ are subdivided in the same way as A_{tt}; i.e., its off-diagonal blocks are again restricted to rank-1 matrices while its diagonal blocks are subdivided and so on.

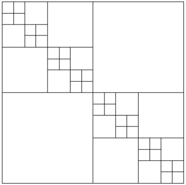


Figure: Partition for $p = 4$

Let N_p^{st} denote the amount of storage which is required to hold such a matrix. Since the off-diagonal blocks in level p require $4 \cdot 2^{p-1} = 2^{p+1}$ units of storage, we find the recursive relation $N_p^{\mathrm{st}} = 2N_{p-1}^{\mathrm{st}} + 2^{p+1}$ with $N_0^{\mathrm{st}} = 1$. Resolving this recursion, we obtain that

$$N_p^{\mathrm{st}} = (2p+1)2^p = 2n\log_2 n + n.$$

Using the blocking (9), the matrix-vector product with $x \in C^I$ can be computed recursively by

$$A_{tt}x_t = \begin{bmatrix} A_{t_1t_1}x_{t_1} + A_{t_1t_2}x_{t_2} \\ A_{t_2t_1}x_{t_1} + A_{t_2t_2}x_{t_2} \end{bmatrix}, \quad \text{where } x_t = \begin{bmatrix} x_{t_1} \\ x_{t_2} \end{bmatrix} \text{ and } x_{t_1} \in \mathbb{C}^{t_1},\, x_{t_2} \in \mathbb{C}^{t_2}.$$

Hence, for Ax we need the results of the products $A_{t_1t_1}x_{t_1}$ and $A_{t_2t_2}x_{t_2}$, which have half the size. For the number N_p^{MV} of operations it holds that

$$N_p^{\mathrm{MV}} = 2N_{p-1}^{\mathrm{MV}} + 2^{p+2} - 2.$$

Multiplying the rank-1 matrices $A_{t_1t_2}$ and $A_{t_2t_1}$ by x_{t_2} and x_{t_1} and adding the results each requires $4 \cdot 2^{p-1} - 1$ operations. With $N_0^{\mathrm{MV}} = 2$ we obtain

$$N_p^{\mathrm{MV}} = p2^{p+2} + 2 = 4n\log_2 n + 2.$$

Cluster trees

The presented partition is too special since non-admissible blocks can only appear on the diagonal. In the following it will be described how hierarchical partitions can be constructed for arbitrary admissibility conditions.
In order to partition the set of matrix indices $I \times J$ hierarchically into sub-blocks, we first need a rule to subdivide the index sets I and J. This leads to the so-called cluster tree (cf. [7]), which contains a hierarchy of partitions.

Definition

A tree $T_I = (V, E)$ with vertices V and edges E is called a **cluster tree** *for a set $I \subset \mathbb{N}$ if the following conditions hold*

(a) *I is the root of T_I;*

(b) *$\varnothing \neq t = \bigcup_{t' \in S(t)} t'$ for all $t \in V$;*

(c) *the degree $\deg t := |S(t)| \geq 2$ of each vertex $t \in V \setminus \mathcal{L}(T_I)$ is bounded from below.*

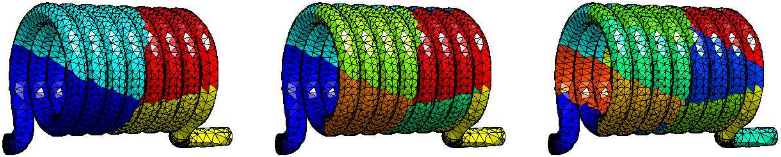
Here, the set of sons $S(t) := \{t' \in V : (t, t') \in E\}$ of $t \in V$ is pairwise disjoint and $\mathcal{L}(T_I) := \{t \in V : S(t) = \varnothing\}$ denotes the set of leaves of T_I.

Condition (b) implies that $t \subset I$ for all $t \in T_I$ and that each level

$$T_I^{(\ell)} := \{t \in T_I : \text{level } t = \ell\}$$

of T_I contains a partition of I.

In the sequel we will identify the set of vertices V with the cluster tree T_I. The purpose of S is to generate subdomains of minimal diameter.



2nd level 3rd level 4th level

Figure: Three levels in a cluster tree.

For practical purposes it is useful to work with clusters having a minimal size of $n_{\min} > 1$ rather than subdividing the clusters until only one index is left.

Since the number of leaves $|\mathcal{L}(T_I)|$ is bounded by $|I|/n_{\min}$ provided $|t| \geq n_{\min}$ for all $t \in T_I$, the following estimate shows that the complexity of storing a cluster tree is still linear. The proof uses the property that each subtree of T_I is a cluster tree.

Lemma

Let $q := \min_{t \in T_I \setminus \mathcal{L}(T_I)} \deg t \geq 2$. *Then for the number of vertices in* T_I *it holds that*

$$|T_I| \leq \frac{q|\mathcal{L}(T_I)| - 1}{q-1} \leq 2|\mathcal{L}(T_I)| - 1. \tag{10}$$

Proof. We cut down the tree T vertex by vertex starting from the leaves of $T \setminus \mathcal{L}(T)$ in k steps until only the root is left. Let T_ℓ denote the tree after ℓ steps and q_ℓ the degree of the ℓth vertex. Then $|T_{\ell+1}| = |T_\ell| - q_\ell$ and $|\mathcal{L}(T_{\ell+1})| = |\mathcal{L}(T_\ell)| - q_\ell + 1$. After k steps $|T_k| = 1 = |\mathcal{L}(T_k)|$, where

$$|T_k| = |T| - \sum_{\ell=1}^{k-1} q_\ell \quad \text{and} \quad |\mathcal{L}(T_k)| = |\mathcal{L}(T)| - \sum_{\ell=1}^{k-1}(q_\ell - 1).$$

Hence, $|T| = |\mathcal{L}(T)| + k - 1$ and from $q_k \geq q$ it follows that $k(q-1) \leq |\mathcal{L}(T)| + q - 2$.
■

The number of vertices in T_I is always linear. In order to guarantee a logarithmic depth of T_I we have to ensure that each subdivision by S produces clusters of comparable cardinality.

Definition

A tree T_I is called **balanced** *if*

$$R := \min_{t \in T_I \setminus \mathcal{L}(T_I)} \{|t_1|/|t_2|,\ t_1, t_2 \in S(t)\}$$

is bounded independently of $|I|$ by a positive constant from below.

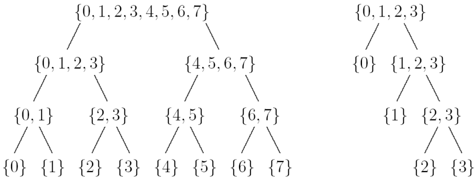


Figure: A balanced and an unbalanced tree.

By $L(T_I) := \max_{t \in T_I} \text{level}\, t + 1$ we denote the **depth** of the cluster tree T_I. The depth of balanced cluster trees depends logarithmically on $|I|$.

Lemma

Let T_I be a balanced cluster tree. Then for the depth of T_I it holds that

$$L(T_I) \le \log_{1+R}(|I|/n_{\min}) + 1 \sim \log |I| \tag{11}$$

and $|t| \le |I|(1+R)^{-\ell}$, where ℓ denotes the level of $t \in T_I$.

Proof. For $t \in T_I \setminus \mathcal{L}(T_I)$ and $t' \in S(t)$ we observe that

$$\frac{|t|}{|t'|} = \frac{|t'| + \sum_{t' \neq s \in S(t)} |s|}{|t'|} = 1 + \sum_{t' \neq s \in S(t)} \frac{|s|}{|t'|} \ge 1 + (|S(t)| - 1)R \ge 1 + R.$$

Let $e_1, \ldots, e_{L-1}$ be a sequence of edges from the root $v_1 := I$ to a deepest vertex v_L in T_I. Furthermore, let $v_2, \ldots, v_{L-1}$ be the intermediate vertices. Then from

$$(1+R)|v_{\ell+1}| \le |v_\ell|, \quad \ell = 1, \ldots, L-1,$$

we obtain that

$$(1+R)^{L-1}|v_L| \le |I|,$$

which gives $(L-1)\log(1+R) \le \log(|I|/|v_L|) \le \log(|I|/n_{\min})$. ■

The following lemma will be helpful for many of the complexity estimates.

Lemma

Let T_I be a cluster tree for I, then

$$\sum_{t\in T_I} |t| \le L(T_I)|I| \quad \textit{and} \quad \sum_{t\in T_I} |t|\log|t| \le L(T_I)|I|\log|I|. \tag{12}$$

Proof. Each of the $L(T_I)$ levels in T_I is made of disjoint subsets of I. The second estimate follows from $\log|t| \le \log|I|$ for $t \in T_I$. ■

For each vertex t in T_I its indices have to be stored. It is desirable that clusters are contiguous. For this purpose, the set I should be reordered. If t is to be subdivided into t_1 and t_2, we rearrange the indices in t so that $\max t_1 \le \min t_2$. This can be done during the construction of T_I. In this case, a cluster t can be represented by its minimal and maximal index. With this simplification, T_I requires $|T_I| \sim |I|$ units of storage even for unbalanced trees. The permutation requires additional $|I|$ units of storage. Note that due to the hierarchical character of cluster trees, this reordering does not change the previously generated clusters.

Construction of cluster trees

In this section we will concentrate on how a cluster tree T_I is constructed from an index set $I \subset \mathbb{N}$ such that the diameters of X_t are as small as possible.
We assume that X_i, $i \in I$, are **quasi-uniform**, i.e., there is a constant $c_U > 0$ such that

$$\max_{i \in I} \mu(X_i) \leq c_U \min_{i \in I} \mu(X_i).$$

The expression $\mu(M)$ denotes the m-dimensional measure of an m-dimensional manifold $M \subset \mathbb{R}^d$. We assume that the computational domain Ω is an m-dimensional manifold, i.e., there is a constant $c_\Omega > 0$ such that for all $z \in \Omega$

$$\mu(\Omega \cap B_r(z)) \leq c_\Omega r^m \quad \text{for all } r > 0. \tag{13}$$

In addition, we assume that only a bounded number of sets X_i overlap; i.e., there is $\nu \in \mathbb{N}$ such that

$$|\{i \in I : \exists j \in J \text{ such that } \operatorname{int} X_i \cap \operatorname{int} X_j \neq \varnothing\}| \leq \nu. \tag{14}$$

The above assumptions are in accordance with usual applications such as finite element discretizations.

We use a clustering strategy which is based on the *principal component analysis* (PCA). In order to be able to apply it, we select arbitrary but fixed points $z_i \in X_i$, $i = 1, \dots, n$, e.g., the centroid of X_i if X_i is polygonal.

A cluster $t \subset I$ is subdivided by the hypersurface through the centroid

$$m_t := \frac{\sum_{i \in t} \mu(X_i) z_i}{\sum_{i \in t} \mu(X_i)}$$

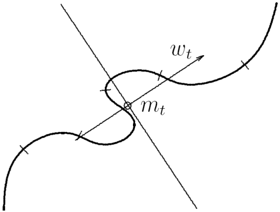
of t with normal w_t, where w_t is the main direction of t.

Definition

A vector $w \in \mathbb{R}^d$, $\|w\|_2 = 1$, satisfying

$$\sum_{i \in t} |w^T(z_i - m_t)|^2 = \max_{\|v\|_2=1} \sum_{i \in t} |v^T(z_i - m_t)|^2$$

is called **main direction** *of the cluster t.*



Note that with the symmetric positive semidefinite **covariance matrix**

$$C_t := \sum_{i\in t} (z_i - m_t)(z_i - m_t)^T \in \mathbb{R}^{d\times d}$$

it holds that

$$\sum_{i\in t} |v^T(z_i - m_t)|^2 = \sum_{i\in t} v^T(z_i - m_t)(z_i - m_t)^T v = v^T C_t v \quad \text{for all } v \in \mathbb{R}^d.$$

Hence, by the variational representation of eigenvalues

$$\max_{\|v\|_2=1} \sum_{i\in t} |v^T(z_i - m_t)|^2 = \max_{\|v\|_2=1} v^T C_t v = \lambda_{\max}(C_t)$$

one observes that the maximum is attained for the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ of C_t.
Using w_t, one possibility is to define the sons $S(t) = \{t_1, t_2\}$ of t by

$$t_1 = \{i \in t : w_t^T(z_i - m_t) > 0\}$$

and $t_2 := t \setminus t_1$. This subdivision generates **geometrically balanced** cluster trees in the sense that there are constants $c_g, c_G > 0$ such that for each level $\ell = 0, \dots, L(T_I) - 1$

$$(\operatorname{diam} X_t)^m \le c_g 2^{-\ell} \quad \text{and} \quad \mu(X_t) \ge 2^{-\ell}/c_G \quad \text{for all } t \in T_I^{(\ell)}. \tag{15}$$

The following lemma shows that geometrically balanced cluster trees are balanced for quasi-uniform grids.

Lemma

Assume that X_i, $i \in I$, are quasi-uniform. Then a geometrically balanced cluster tree is balanced.

Proof. Let $t_1, t_2 \in T_I^{(\ell)}$ be two clusters from the same level ℓ of T_I. From (14), (13), and (15) we see that

$$|t_1| \min_{i\in t_1} \mu(X_i) \le \sum_{i\in t_1} \mu(X_i) \le \nu\mu(X_{t_1}) \le \nu c_\Omega (\operatorname{diam} X_{t_1})^m \le \nu c_\Omega c_g 2^{-\ell}.$$

The last estimate together with

$$|t_2| \max_{i\in t_2} \mu(X_i) \ge \sum_{i\in t_2} \mu(X_i) \ge \mu(X_{t_2}) \ge 2^{-\ell}/c_G$$

leads to

$$\frac{|t_1|}{|t_2|} \le \nu c_\Omega c_g c_G \frac{\max_{i\in t_2} \mu(X_i)}{\min_{i\in t_1} \mu(X_i)} \le \nu c_\Omega c_g c_G c_U,$$

which proves the assertion. ■

Since subdividing $t \subset I$ requires $\mathcal{O}(|t|)$ operations for geometrically balanced clustering, we can see that constructing T_I takes $L(T_I)|I|$ operations.

Block Cluster Trees

Since our aim is to find an admissible partition of $I \times J$, we consider cluster trees $T_{I\times J}$ for $I \times J$, which will be referred to as **block cluster trees**.
Let T_I and T_J be cluster trees for I and J with corresponding mappings S_I and S_J. We will consider block cluster trees $T_{I\times J}$ which are defined by the following mapping $S_{I\times J}$:

$$S_{I\times J}(t\times s) = \begin{cases} \varnothing, & \text{if } t\times s \text{ is admissible or } S_I(t)=\varnothing \text{ or } S_J(s)=\varnothing, \\ S_I(t)\times S_J(s), & \text{else.} \end{cases}$$

The depth $L(T_{I\times J})$ of the tree $T_{I\times J}$ is obviously bounded by the minimum of the depths of the generating cluster trees T_I and T_J; i.e.,

$$L(T_{I\times J}) \le \min\{L(T_I), L(T_J)\}.$$

The leaves of $T_{I\times J}$ constitute an admissible partition $P := \mathcal{L}(T_{I\times J})$.
Usually, a block cluster is built from binary cluster trees; i.e., $\deg t = 2$ for $t \in T_I \setminus \mathcal{L}(T_I)$. In this case, $T_{I\times J}$ is a quadtree. Note that each block $t\times s \in T_{I\times J}$ consists of clusters $t \in T_I$ and $s \in T_J$ of the same level from their respective cluster tree.

The sparsity constant

A measure for the complexity of a partition is the so-called sparsity constant; cf. [4]. The "sparsity" is the maximum number of blocks in the partition that are associated with a given row or column cluster.

Definition

Let T_I and T_J be cluster trees for the index sets I and J and let $T_{I\times J}$ be a block cluster tree for $I \times J$. For a cluster $t \in T_I$ we denote the maximum number of blocks $t \times s \in T_{I\times J}$ by

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

Similarly, for a given cluster $s \in T_J$

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

stands for the maximum number of blocks $t \times s \in T_{I\times J}$. The **sparsity constant** *c_{sp} of a block cluster tree $T_{I\times J}$ is then defined as*

$$c_{\text{sp}}(T_{I\times J}) := \max\left\{\max_{t\in T_I} c_{\text{sp}}^{r}(T_{I\times J}, t), \max_{s\in T_J} c_{\text{sp}}^{c}(T_{I\times J}, s)\right\}.$$

Lemma

Assume that T_I and T_J are geometrically balanced, i.e., (15) *is satisfied. Then $c_{\text{sp}}(T_{I\times J}) \leq 2\nu c_\Omega c_g c_G (2 + 1/\eta)^m$ holds.*

Proof. Let $t \in T_I^{(\ell)}$ with an associated point $z_t \in X_t$. The estimates (14) and (15) guarantee that each neighborhood

$$N_\rho := \{s \in T_J^{(\ell)} : \max_{x \in X_s} |x - z_t| \le \rho\}, \quad \rho > 0,$$

of t contains at most $\nu c_G c_\Omega 2^\ell \rho^m$ clusters s from the same level ℓ in T_J. This follows from

$$|N_\rho| 2^{-\ell}/c_G \le \sum_{s \in N_\rho} \mu(X_s) \le \nu \mu(X_{N_\rho}) \le \nu c_\Omega \rho^m. \tag{16}$$

Let $s \in T_J$ such that $t \times s \in T_{I \times J}$. Furthermore, let t^* and s^* be the father clusters of t and s, respectively. Assume that $\max_{x \in X_s} |x - z_t| \ge \rho_0$, where

$$\begin{aligned}\rho_0 &:= \min\{\operatorname{diam} X_{t^*}, \operatorname{diam} X_{s^*}\}/\eta + \operatorname{diam} X_{t^*} + \operatorname{diam} X_{s^*} \\ &\le (c_g 2^{-(\ell-1)})^{1/m}(2 + 1/\eta).\end{aligned}$$

Then

$$\begin{aligned}\operatorname{dist}(X_{t^*}, X_{s^*}) &\ge \max_{x \in X_s} |x - z_t| - \operatorname{diam} X_{t^*} - \operatorname{diam} X_{s^*} \\ &\ge \min\{\operatorname{diam} X_{t^*}, \operatorname{diam} X_{s^*}\}/\eta\end{aligned}$$

implies that $t^* \times s^*$ is admissible. Thus, $T_{I \times J}$ cannot contain $t \times s$, which is a contradiction. It follows that $\max_{x \in X_s} |x - z_t| < \rho_0 \le (c_g 2^{-(\ell-1)})^{1/m}(2 + 1/\eta)$. From (16) we obtain that

$$c_{\text{sp}}^r = |\{s \in T_J : t \times s \in P\}| \le 2\nu c_\Omega c_g c_G (2 + 1/\eta)^m.$$

Interchanging the roles of t and s, one shows that c_{sp}^c is bounded by the same constant.

■

Many algorithms in the context of hierarchical matrices can be applied blockwise. In this case, the cost of the algorithm is the sum over the costs of each block. Assume that on each block the costs are bounded by a constant $c > 0$. Then

$$\sum_{t\times s\in T_{I\times J}} c \le c \sum_{t\in T_I} |\{s\subset J : t\times s\in T_{I\times J}\}| \le c c_{\text{sp}}^r |T_I| \le c c_{\text{sp}} |T_I|. \tag{17}$$

By interchanging the roles of t and s, we also obtain $\sum_{t\times s\in T_{I\times J}} c \le c c_{\text{sp}}|T_J|$.
If the cost of the algorithm on each block $t\times s\in P$ is bounded by $c(|t|+|s|)$, then the overall cost can be estimated as

$$\begin{aligned}\sum_{t\times s\in T_{I\times J}} c(|t|+|s|) &= c\sum_{t\in T_I'} \sum_{\{s\in T_J : t\times s\in T_{I\times J}\}} |t| + c\sum_{s\in T_J'} \sum_{\{t\in T_I : t\times s\in T_{I\times J}\}} |s| && \text{(18a)}\\ &\le c c_{\text{sp}}\left(\sum_{t\in T_I'}|t| + \sum_{s\in T_J'}|s|\right) \le c c_{\text{sp}} L(T_{I\times J})[|I|+|J|] && \text{(18b)}\\ &\le c c_{\text{sp}}[L(T_I)|I| + L(T_J)|J|] && \text{(18c)}\end{aligned}$$

due to (12). Here, T_I' and T_J' denote the subtrees of T_I and T_J, respectively, which are actually used to construct $T_{I\times J}$; i.e.,

$$T_I' := \{t\in T_I : \text{there is } t'\subset t \text{ and } s'\in T_J \text{ such that } t'\times s'\in T_{I\times J}\}.$$

Lemma

Let T_I and T_J be cluster trees for the index sets I and J, respectively. For the number of blocks in a partition $\mathcal{L}(T_{I\times J})$ and for the number of vertices in $T_{I\times J}$ it holds that

$$|\mathcal{L}(T_{I\times J})| \leq |T_{I\times J}| \leq 2c_{\text{sp}} \min\left\{|\mathcal{L}(T_I)|, |\mathcal{L}(T_J)|\right\}.$$

If $|t| \geq n_{\min}$ for all $t \in T_I \cup T_J$, then $|T_{I\times J}| \leq 2c_{\text{sp}} \min\left\{|I|, |J|\right\}/n_{\min}$.

Proof. Estimate (17) applied to $|T_{I\times J}| = \sum_{t\times s\in T_{I\times J}} 1$ gives $|T_{I\times J}| \leq c_{\text{sp}} \min\{|T_I|, |T_J|\}$. Equation (10) states that $|T_I| \leq 2|\mathcal{L}(T_I)|$ and $|T_J| \leq 2|\mathcal{L}(T_J)|$. ■

Lemma

Let T_I and T_J be balanced cluster trees. Taking into account requirement (b) on the admissibility condition, the number of operations for constructing the block cluster tree is of the order $c_{\text{sp}}[|I|\log|I| + |J|\log|J|]$.

Proof. Use (18) and (11). ■

The time required to compute an admissible matrix partition can be neglected compared with the rest of the computation.

The Set of Hierarchical Matrices

Definition

The set of **hierarchical matrices** *on the block cluster tree $T_{I\times J}$ with admissible partition $P := \mathcal{L}(T_{I\times J})$ and blockwise rank k is defined as*

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

For the sake of brevity, elements from $\mathcal{H}(T_{I\times J}, k)$ will be called $\mathcal{H}$-matrices.

Remark

For an efficient treatment of admissible blocks the outer-product representation should be used. Additionally, it is advisable not to use the maximum rank k but the actual rank of the respective block as the number of rows and columns. Storing non-admissible blocks entrywise will increase the efficiency.

We have already seen that the storage requirements for an admissible block $b = t \times s \in \mathcal{L}(T_{I\times J})$ of $A \in \mathcal{H}(T_{I\times J}, k)$ are

$$N_{\mathrm{st}}(A_b) = k(|t| + |s|).$$

A non-admissible block A_b, $b \in P$, is stored entrywise and thus requires $|t||s|$ units of storage. Since $\min\{|t|, |s|\} \leq n_{\min}$ we have

$$|t||s| = \min\{|t|, |s|\} \max\{|t|, |s|\} \leq n_{\min}(|t| + |s|). \tag{19}$$

Hence, for storing A_b, $b \in P$, at most $\max\{k, n_{\min}\}(|t| + |s|)$ units of storage are required. Using (18), we obtain the following theorem.

Theorem

Let c_{sp} be the sparsity constant of the partition P. The storage requirements N_{st} for $A \in \mathcal{H}(T_{I\times J}, k)$ are bounded by

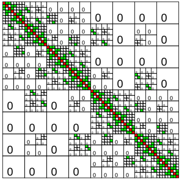
$$N_{\mathrm{st}}(A) \leq c_{\mathrm{sp}} \max\{k, n_{\min}\}[L(T_I)|I| + L(T_J)|J|].$$

If T_I and T_J are balanced cluster trees, we have

$$N_{\mathrm{st}}(A) \sim \max\{k, n_{\min}\}[|I| \log |I| + |J| \log |J|].$$

Sparse $\mathcal{H}$-matrices

Although $\mathcal{H}$-matrices are primarily aiming at dense matrices, sparse matrices A which vanish on admissible blocks are also in $\mathcal{H}(T_{I\times I}, n_{\min})$. Since the size of one of the clusters corresponding to non-admissible blocks is less than or equal to $n_{\min}$, the rank of each block A_b does not exceed $n_{\min}$.



The following lemma shows that the storage requirements are actually linear.

Lemma

Storing FE matrices A as an $\mathcal{H}$-matrix requires $\mathcal{O}(n)$ units of storage.

Proof. Since A vanishes on admissible blocks, we only have to estimate the number of non-admissible blocks. Let $t \in T_I^{(\ell)}$ be a cluster from the ℓth level of T_I. The number of elements of the set

$$N(t) := \left\{ s \in T_J^{(\ell)} : \eta \operatorname{dist}(X_t, X_s) \le \min\{\operatorname{diam} X_t, \operatorname{diam} X_s\} \right\}$$

is bounded by a constant since due to (15)

$$\begin{aligned} |N(t)| 2^{-\ell} / c_G &\le \sum_{s \in N(t)} \mu(X_s) \le \nu \mu(X_{N(t)}) \\ &\le \nu c_\Omega (\operatorname{diam} X_t + \eta^{-1} \min\{\operatorname{diam} X_t, \operatorname{diam} X_s\})^d \\ &= \nu c_\Omega (1 + 1/\eta)^d c_g 2^{-\ell} \end{aligned}$$

gives $|N(t)| \le \nu c_g c_G c_\Omega (1 + 1/\eta)^d$. Since there are only $|T_I| \sim |I|$ cluster $t \in T_I$, we obtain the desired result. ■

Matrix-Vector Multiplication

Multiplying an $\mathcal{H}$-matrix $A \in \mathcal{H}(T_{I\times J}, k)$ or its Hermitian transpose A^H by a vector x can be done blockwise:

$$Ax = \sum_{t\times s\in P} A_{ts}x_s \quad \text{and} \quad A^H x = \sum_{t\times s\in P} (A_{ts})^H x_t.$$

Since each admissible block $t \times s$ has the outer product representation $A_{ts} = UV^H$, $U \in \mathbb{C}^{t\times k}$, $V \in \mathbb{C}^{s\times k}$, at most $2k(|t| + |s|)$ operations are required to compute the matrix-vector products $A_{ts}x_s = UV^H x_s$ and $(A_{ts})^H x_t = VU^H x_t$. If $t \times s$ is non-admissible, then A_{ts} is stored entrywise and $\min\{|t|, |s|\} \le n_{\min}$. As in (19) we see that in this case $2|t||s| \le 2n_{\min}(|t| + |s|)$ arithmetic operations are required.

Theorem

For the number of operations N_{MV} required for one matrix-vector multiplication Ax of $A \in \mathcal{H}(T_{I\times J}, k)$ by a vector $x \in \mathbb{C}^J$ it holds that

$$N_{\text{MV}}(A) \le 2c_{\text{sp}} \max\{k, n_{\min}\}[L(T_I)|I| + L(T_J)|J|].$$

If T_I and T_J are balanced cluster trees, we have

$$N_{\text{MV}}(A) \sim \max\{k, n_{\min}\}[|I| \log |I| + |J| \log |J|].$$

Hence, $\mathcal{H}$-matrices are well suited for iterative schemes such as Krylov subspace methods which the matrix enters only through the matrix-vector product.

Blockwise and global norms

From the analysis we will usually obtain estimates on each of the blocks b of a partition P. However, such estimates are finally required for the whole matrix. If we are interested in the Frobenius norm, blockwise estimates directly translate to global estimates using

$$\|A\|_F^2 = \sum_{b\in P} \|A_b\|_F^2.$$

For the spectral norm the situation is a bit more difficult. We can however exploit the structure of the partition P together with the following lemma.

Lemma

Consider the following $r \times r$ block matrix

$$A = \begin{bmatrix} A_{11} & \dots & A_{1r} \\ \vdots & & \vdots \\ A_{r1} & \dots & A_{rr} \end{bmatrix}$$

with $A_{ij} \in \mathbb{C}^{m_i \times n_j}$, $i, j = 1, \dots, r$. Then it holds that

$$\max_{i,j=1,\dots,r} \|A_{ij}\|_2 \le \|A\|_2 \le \left(\max_{i=1,\dots,r} \sum_{j=1}^{r} \|A_{ij}\|_2 \right)^{1/2} \left(\max_{j=1,\dots,r} \sum_{i=1}^{r} \|A_{ij}\|_2 \right)^{1/2}.$$

Theorem

Let P be the leaves of a block cluster tree $T_{I\times J}$. Then for $A, B \in \mathcal{H}(T_{I\times J}, k)$ it holds that

(a) $\max_{b\in P} \|A_b\|_2 \le \|A\|_2 \le c_{\text{sp}} L(T_{I\times J}) \max_{b\in P} \|A_b\|_2$

(b) $\|A\|_2 \le c_{\text{sp}} L(T_{I\times J}) \|B\|_2$ *provided* $\max_{b\in P} \|A_b\|_2 \le \max_{b\in P} \|B_b\|_2$.

Proof. Let A_ℓ denote the part of A which corresponds to the blocks of P from the ℓth level of $T_{I\times J}$; i.e.,

$$(A_\ell)_b = \begin{cases} A_b, & b \in T_{I\times J}^{(\ell)} \cap P, \\ 0, & \text{else.} \end{cases}$$

Since A_ℓ has tensor structure with at most c_{sp} per block row or block column, the last lemma gives $\|A_\ell\|_2 \le c_{\text{sp}} \max_{b\in T_{I\times J}^{(\ell)}\cap P} \|A_b\|_2$, such that

$$\|A\|_2 \le \sum_{\ell=1}^{L(T_{I\times J})} \|A_{\ell-1}\|_2 \le c_{\text{sp}} \sum_{\ell=1}^{L(T_{I\times J})} \max_{b\in T_{I\times J}^{(\ell-1)}\cap P} \|A_b\| \le c_{\text{sp}} L(T_{I\times J}) \max_{b\in P} \|A_b\|_2.$$

The estimate

$$\max_{b\in P} \|A_b\|_2 \le \max_{b\in P} \|B_b\|_2 \le \|B\|_2$$

gives the second part of the assertion. ■

Adding $\mathcal{H}$-Matrices

Since $\mathcal{H}(T_{I\times J},k)$ is not a linear space, we have to approximate the sum $A+B$ by a matrix $S\in\mathcal{H}(T_{I\times J},k)$ if we want to avoid that the rank and hence the complexity grows with each addition.

- use rounded addition on each admissible block;
- on non-admissible block, the usual addition is employed.

Since the rounded addition gives a blockwise best approximation, S is a best approximation in the Frobenius norm

$$\|A+B-S\|_F \le \|A+B-M\|_F \quad \text{for all } M\in\mathcal{H}(T_{I\times J},k).$$

Using the last theorem, this estimate for the spectral norm reads

$$\|A+B-S\|_2 \le c_{\text{sp}}L(T_{I\times J})\|A+B-M\|_2 \quad \text{for all } M\in\mathcal{H}(T_{I\times J},k).$$

The following bound on the number of arithmetic operations results from the complexity of the rounded addition, (17), and (18).

Theorem

Let $A,B\in\mathcal{H}(T_{I\times J},k)$. A matrix $S\in\mathcal{H}(T_{I\times J},k)$ satisfying the above error estimates can be computed with at most

$$13c_{\text{sp}}k^2[L(T_I)|I|+L(T_J)|J|]+176c_{\text{sp}}k^3\min\{|T_I|,|T_J|\}$$

arithmetic operations.

Preserving Positivity

If the smallest eigenvalue is close to the origin compared with the rounding accuracy, it may happen that the rounded result becomes indefinite although it should be positive definite in exact arithmetic.

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

$$\max\{\|E\|_2, \|F\|_2\} \leq \sqrt{\varepsilon}$$

is associated with $t \times s$; i.e.,

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

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

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

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

Since

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

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

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

Note that adding EE^H to A_{tt} and FF^H to A_{ss} leads to a rounding error which in turn has to be added to the diagonal sub-blocks of $t \times t$ and $s \times s$ in order to preserve positivity. For this purpose, we add a positive semi-definite matrix. Let t_1 and t_2 be the sons of t and let s_1 and s_2 be the sons of s. If we define

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

the problem of adding EE^H to A_{tt} is reduced to adding $2E_{t_1} E_{t_1}^H$ to $A_{t_1 t_1}$ and $2E_{t_2} E_{t_2}^H$ to $A_{t_2 t_2}$. Applying this idea recursively, adding EE^H to A_{tt} can finally be done by adding a multiple of $E_{t'} E_{t'}^H$ to the dense matrix block $A_{t' t'}$ for each leaf t' in T_I from the set of descendants of t.

We obtain the following two algorithms addsym_stab and addsym_diag.

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

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

```
procedure addsym_diag($t, E, \mathrm{var}\ A$)
if $t \times t$ is a leaf then
    add $EE^H$ to $A_{tt}$ without approximation;
else
    addsym_diag($t_1, \sqrt{2}E_{t_1}, A$);
    addsym_diag($t_2, \sqrt{2}E_{t_2}, A$);
endif
```

Theorem

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

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

At most $(\max\{k^2, n_{\min}\} + n_{\min} k)L(T_I)|I|$ operations are needed to construct $\tilde{S}_{\mathcal{H}}$.

Multiplying $\mathcal{H}$-Matrices

Let $A \in \mathcal{H}(T_{I\times J}, k_A)$ and $B \in \mathcal{H}(T_{J\times K}, k_B)$ be two hierarchical matrices. We compute an approximation C of the product in $\mathcal{H}(T_{I\times K}, \tilde{k})$.

Assume that A and B are subdivided according to their block cluster trees $T_{I\times J}$ and $T_{J\times K}$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

Then AB has the following block structure

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

Assume that the products $A_{ik}B_{kj}$, $i, j, k = 1, 2$, each of which has half the size of AB, have been computed.

- Round the sums $A_{i1}B_{1j} + A_{i2}B_{2j}$ to rank-$\tilde{k}$ matrices R_{ij};
- if C has a 2×2 block structure in $T_{I\times K}$, then $C_{ij} := C_{ij} + R_{ij}$, $i, j = 1, 2$.
- else: agglomerate
$$\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$$
to a single rank-$\tilde{k}$ matrix and add it to C.

The complexity of this multiplication was shown to be of the order $k^2 L^2(T_I)|I| + k^3|I|$ for $I = J$.

Hierarchical Inversion

Assume that a block A_{tt} is subdivided into sub-blocks in the following way:

$$A_{tt} = \begin{bmatrix} A_{t_1 t_1} & A_{t_1 t_2} \\ A_{t_2 t_1} & A_{t_2 t_2} \end{bmatrix},$$

where t_1 and t_2 denote the sons of t in T_I. For the exact inverse of A_{tt} it holds

$$A_{tt}^{-1} = \begin{bmatrix} A_{t_1 t_1}^{-1} + A_{t_1 t_1}^{-1} A_{t_1 t_2} S^{-1} A_{t_2 t_1} A_{t_1 t_1}^{-1} & -A_{t_1 t_1}^{-1} A_{t_1 t_2} S^{-1} \\ -S^{-1} A_{t_2 t_1} A_{t_1 t_1}^{-1} & S^{-1} \end{bmatrix},$$

where S denotes the Schur complement $S := A_{t_2 t_2} - A_{t_2 t_1} A_{t_1 t_1}^{-1} A_{t_1 t_2}$ of $A_{t_1 t_1}$ in A. Let $T \in \mathcal{H}(T_{I\times I}, k)$ which together with C is initialized to zero.

```
procedure invertH($t, A, \mathtt{var}\ C$)
if $t \in \mathcal{L}(T_I)$ then $C_{tt} := A_{tt}^{-1}$ is the usual inverse
else
    invertH($t_1, A, C$)
    $T_{t_1 t_2} = T_{t_1 t_2} - C_{t_1 t_1} A_{t_1 t_2}$
    $T_{t_2 t_1} = T_{t_2 t_1} - A_{t_2 t_1} C_{t_1 t_1}$
    $A_{t_2 t_2} = A_{t_2 t_2} + A_{t_2 t_1} T_{t_1 t_2}$
    invertH($t_2, A, C$)
    $C_{t_1 t_2} = C_{t_1 t_2} + T_{t_1 t_2} C_{t_2 t_2}$
    $C_{t_2 t_1} = C_{t_2 t_1} + C_{t_2 t_2} T_{t_2 t_1}$
    $C_{t_1 t_1} = C_{t_1 t_1} + T_{t_1 t_2} C_{t_2 t_1}$
```

The complexity of the computation of the $\mathcal{H}$-inverse is determined by the cost of the $\mathcal{H}$-matrix multiplication.

Hierarchical LU Decomposition

Although the hierarchical inversion has almost linear complexity, the following hierarchical LU decomposition provides a significantly more efficient alternative. .
To define the $\mathcal{H}$-LU decomposition, we exploit the hierarchical block structure of a block A_{tt}, $t \in T_I \setminus \mathcal{L}(T_I)$:

$$A_{tt} = \begin{bmatrix} A_{t_1 t_1} & A_{t_1 t_2} \\ A_{t_2 t_1} & A_{t_2 t_2} \end{bmatrix} = \begin{bmatrix} L_{t_1 t_1} & \\ L_{t_2 t_1} & L_{t_2 t_2} \end{bmatrix} \begin{bmatrix} U_{t_1 t_1} & U_{t_1 t_2} \\ & U_{t_2 t_2} \end{bmatrix},$$

where $t_1, t_2 \in T_I$ denote the sons of t in T_I. Hence, the LU decomposition of a block A_{tt} is reduced to the following four problems on the sons of $t \times t$:

(a) Compute $L_{t_1 t_1}$ and $U_{t_1 t_1}$ from the LU decomposition $L_{t_1 t_1} U_{t_1 t_1} = A_{t_1 t_1}$;

(b) Compute $U_{t_1 t_2}$ from $L_{t_1 t_1} U_{t_1 t_2} = A_{t_1 t_2}$;

(c) Compute $L_{t_2 t_1}$ from $L_{t_2 t_1} U_{t_1 t_1} = A_{t_2 t_1}$;

(d) Compute $L_{t_2 t_2}$ and $U_{t_2 t_2}$ from the LU decomposition $L_{t_2 t_2} U_{t_2 t_2} = A_{t_2 t_2} - L_{t_2 t_1} U_{t_1 t_2}$.

If a block $t \times t \in \mathcal{L}(T_{I \times I})$ is a leaf, the usual pivoted LU decomposition is employed. For (a) and (d) two LU decompositions of half the size have to be computed.

In order to solve (b), i.e., solve a problem of the structure $L_{tt}B_{ts} = A_{ts}$ for B_{ts}, where L_{tt} is a lower triangular matrix and $t \times s \in T_{I\times I}$, we use a recursive block forward substitution: If the block $t \times s$ is not a leaf in $T_{I\times I}$, from the subdivision of the blocks A_{ts}, B_{ts} and L_{tt} into their sub-blocks

$$\begin{bmatrix} L_{t_1 t_1} & \\ L_{t_2 t_1} & L_{t_2 t_2} \end{bmatrix} \begin{bmatrix} B_{t_1 s_1} & B_{t_1 s_2} \\ B_{t_2 s_1} & B_{t_2 s_2} \end{bmatrix} = \begin{bmatrix} A_{t_1 s_1} & A_{t_1 s_2} \\ A_{t_2 s_1} & A_{t_2 s_2} \end{bmatrix}$$

one observes that B_{ts} can be found from the following equations

$$\begin{aligned} L_{t_1 t_1} B_{t_1 s_1} &= A_{t_1 s_1} \\ L_{t_1 t_1} B_{t_1 s_2} &= A_{t_1 s_2} \\ L_{t_2 t_2} B_{t_2 s_1} &= A_{t_2 s_1} - L_{t_2 t_1} B_{t_1 s_1} \\ L_{t_2 t_2} B_{t_2 s_2} &= A_{t_2 s_2} - L_{t_2 t_1} B_{t_1 s_2}, \end{aligned}$$

which are again of type (b). If on the other hand $t \times s$ is a leaf, the usual forward substitution is applied. Similarly, one can solve (c) by recursive block backward substitution.

The complexity of the above recursions is determined by the complexity of the hierarchical matrix multiplication, which can be estimated as $\mathcal{O}(k^2|I|\log^2|I|)$ for two matrices from $\mathcal{H}(T_{I\times I}, k)$.

In the case of positive definite matrices A it is possible to define an $\mathcal{H}$-version of the Cholesky decomposition of a block A_{tt}, $t \in T_I \setminus \mathcal{L}(T_I)$:

$$A_{tt} = \begin{bmatrix} A_{t_1 t_1} & A_{t_1 t_2} \\ A_{t_1 t_2}^H & A_{t_2 t_2} \end{bmatrix} = \begin{bmatrix} L_{t_1 t_1} & \\ L_{t_2 t_1} & L_{t_2 t_2} \end{bmatrix} \begin{bmatrix} L_{t_1 t_1} & \\ L_{t_2 t_1} & L_{t_2 t_2} \end{bmatrix}^H .$$

This factorization is recursively computed by

$$\begin{aligned} L_{t_1 t_1} L_{t_1 t_1}^H &= A_{t_1 t_1} \\ L_{t_1 t_1} L_{t_2 t_1}^H &= A_{t_1 t_2} \\ L_{t_2 t_2} L_{t_2 t_2}^H &= A_{t_2 t_2} - L_{t_2 t_1} L_{t_2 t_1}^H \end{aligned}$$

using the usual Cholesky decomposition on the leaves of $T_{I \times I}$. The second equation $L_{t_1 t_1} L_{t_2 t_1}^H = A_{t_1 t_2}$ is solved for $L_{t_2 t_1}$ in a similar way as $U_{t_1 t_2}$ has previously been obtained in the LU decomposition.

Once A has been decomposed, the solution of $Ax = b$ can be found by forward/backward substitution: $L_{\mathcal{H}} y = b$ and $U_{\mathcal{H}} x = y$. Since $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ are $\mathcal{H}$-matrices, y_t, $t \in T_I \setminus \mathcal{L}(T_I)$, can be computed recursively by solving the following systems for y_{t_1} and y_{t_2}

$$L_{t_1 t_1} y_{t_1} = b_{t_1} \quad \text{and} \quad L_{t_2 t_2} y_{t_2} = b_{t_2} - L_{t_2 t_1} y_{t_1}.$$

If $t \in \mathcal{L}(T_I)$ is a leaf, a usual triangular solver is used. The backward substitution can be done analogously. These substitutions are exact and their complexity is determined by the complexity of the hierarchical matrix-vector multiplication, which is $\mathcal{O}(k|I| \log |I|)$ for multiplying an $\mathcal{H}(T_{I \times I}, k)$-matrix by a vector.

NOTE:

- We have assumed that the blockwise rank k required for a given approximation accuracy stays bounded during the computation;
- For elliptic problems the required rank can be proved to depend logarithmically on n and on the accuracy ε.

Adaptive Cross Approximation (Computation of Approximants for BEM)

We assume that a partition has been generated. Blocks $b \in P$ which do not satisfy (7) are generated and stored without approximation. All other blocks $b \in P$ satisfy (7) and can be treated independently from each other. Therefore, in the rest of this section we focus on a single block of a discrete integral operator

$$A_{ts} = \Lambda_{1,t}\mathcal{A}\Lambda_{2,s}^{*}$$

which is identified with $A \in \mathbb{R}^{m\times n}$.
The idea of the algorithm is as follows. Starting from $R_0 := A$, find a nonzero pivot in R_k, say (i_k, j_k), and subtract a scaled outer product of the i_kth row and the j_kth column:

$$R_{k+1} := R_k - [(R_k)_{i_k j_k}]^{-1}(R_k)_{1:m,j_k}(R_k)_{i_k,1:n}, \tag{20}$$

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

$$|(R_{k-1})_{i_k j_k}| = \max_{j=1,\dots,n} |(R_{k-1})_{i_k j}|.$$

The choice of i_k is a bit more delicate.

Example

We apply two steps of equation (20) to the following matrix R_0. The bold entries are the chosen pivots.

$$R_0 = \begin{bmatrix} 0.431 & 0.354 & \mathbf{0.582} & 0.417 & 0.455 \\ 0.491 & 0.396 & 0.674 & 0.449 & 0.427 \\ 0.446 & 0.358 & 0.583 & 0.413 & 0.441 \\ 0.380 & 0.328 & 0.557 & 0.372 & 0.349 \\ 0.412 & 0.340 & 0.516 & 0.375 & 0.370 \end{bmatrix} \xrightarrow{\substack{i_1=1\\ j_1=3}} \frac{1}{0.582} \begin{bmatrix} 0.582 \\ 0.674 \\ 0.583 \\ 0.557 \\ 0.516 \end{bmatrix} \begin{bmatrix} 0.431 \\ 0.354 \\ 0.582 \\ 0.417 \\ 0.455 \end{bmatrix}^T ,$$

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -0.008 & -0.014 & 0 & -0.033 & \mathbf{-0.100} \\ 0.014 & 0.003 & 0 & -0.004 & -0.014 \\ -0.032 & -0.011 & 0 & -0.026 & -0.087 \\ 0.029 & 0.025 & 0 & 0.005 & -0.034 \end{bmatrix} \xrightarrow{\substack{i_2=2\\ j_2=5}} \frac{1}{-0.1} \begin{bmatrix} 0 \\ -0.100 \\ -0.014 \\ -0.087 \\ -0.034 \end{bmatrix} \begin{bmatrix} -0.008 \\ -0.014 \\ 0 \\ -0.033 \\ -0.100 \end{bmatrix}^T ,$$

$$R_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{0.016} & 0.005 & 0 & 0.000 & 0 \\ -0.02 & 0.001 & 0 & 0.002 & 0 \\ 0.032 & 0.030 & 0 & 0.017 & 0 \end{bmatrix} \xrightarrow{\substack{i_3=3\\ j_3=1}} \frac{1}{0.016} \begin{bmatrix} 0 \\ 0 \\ 0.016 \\ -0.02 \\ 0.032 \end{bmatrix} \begin{bmatrix} 0.016 \\ 0.005 \\ 0 \\ 0.000 \\ 0 \end{bmatrix}^T$$

Apparently, the size of the entries decreases from step to step.

Since in the kth step only the entries in the j_kth column and the i_kth row of R_k are used to compute R_{k+1}, there is no need to build the whole matrix R_k.

```
Let $k = 1$;  $Z = \varnothing$;
repeat
    find $i_k$ as described later on
    $\tilde{v}_k := a_{i_k,1:n}$
    for $\ell = 1, \dots, k-1$ do $\tilde{v}_k := \tilde{v}_k - (u_\ell)_{i_k} v_\ell$
    $Z := Z \cup \{i_k\}$
    if $\tilde{v}_k$ does not vanish then
        $j_k := \text{argmax}_{j=1,\dots,n} |(\tilde{v}_k)_j|; \quad v_k := (\tilde{v}_k)_{j_k}^{-1} \tilde{v}_k$
        $u_k := a_{1:m,j_k}$
        for $\ell = 1, \dots, k-1$ do $u_k := u_k - (v_\ell)_{j_k} u_\ell$.
        $k := k + 1$
until the stopping criterion (21) is fulfilled or $Z = \{1, \dots, m\}$
```

The matrix $S_k := \sum_{\ell=1}^{k} u_\ell v_\ell^T$ will be used as an approximation of $A = S_k + R_k$. Obviously, the rank of S_k is bounded by k.

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

$$\|u_{k+1}\|_2 \, \|v_{k+1}\|_2 \leq \frac{\varepsilon(1-\eta)}{1+\varepsilon} \, \|S_k\|_F \tag{21}$$

can be used as a stopping criterion. Assume that $\|R_{k+1}\|_F \leq \eta \|R_k\|_F$ with η from (7), then

$$\|R_k\|_F \leq \|R_{k+1}\|_F + \|u_{k+1} v_{k+1}^T\|_F \leq \eta \|R_k\|_F + \|u_{k+1}\|_2 \, \|v_{k+1}\|_2.$$

Hence,

$$\|R_k\|_F \leq \frac{1}{1-\eta} \, \|u_{k+1}\|_2 \, \|v_{k+1}\|_2 \leq \frac{\varepsilon}{1+\varepsilon} \, \|S_k\|_F \leq \frac{\varepsilon}{1+\varepsilon} \, (\|A\|_F + \|R_k\|_F).$$

From the last estimate we obtain $\|R_k\|_F \leq \varepsilon \|A\|_F$; i.e., condition (21) guarantees a relative approximation error ε.

Due to (2), the Frobenius norm of S_k can be computed with $\mathcal{O}(k^2(m+n))$ complexity. Therefore, the amount of numerical work required by Algorithm 7.1 is of the order $|Z|^2(m+n)$, which leads to an overall logarithmic-linear complexity due to (18).

Remark

If the costs for generating the matrix entries dominate the algebraic transformations of Algorithm 7.1, then its complexity scales like $|Z|(m+n)$.

What remains is to estimate the remainder R_k. For this purpose, the entries of R_k will be estimated by the approximation error

$$F_{ts}^{\Xi} := \max_{j\in s} \inf_{p\in \operatorname{span}\Xi} \|\mathcal{A}\Lambda_{2,j}^* - p\|_{\infty,Y_t}$$

in an arbitrary system of functions $\Xi := \{\xi_1, \dots, \xi_{k'}\}$ with $\xi_1 = 1$. Note that $\mathcal{A}\Lambda_{2,j}^*$ is an asymptotically smooth function since $\operatorname{supp}\Lambda_{2,j}^* = X_j$.

For the linear operators $\Lambda_{1,i}$, $i = 1, \dots, k'$, corresponding to the first k' rows in A we assume that

$$\det\,[\Lambda_{1,i}\xi_j]_{i,j=1,\dots,k'} \neq 0,$$

which can be guaranteed by the choice of pivoting rows i_k.

Theorem

Then for $i = 1, \dots, m$ and $j = 1, \dots, n$ it holds that

$$|(R_k)_{ij}| \le c(1 + \|\mathfrak{I}_{k'}^{\Xi}\|)(1 + 2^k)\|\psi_i\|_{L^1} F_{ts}^{\Xi}. \tag{22}$$

The similarity of ACA and the *LU* factorization can be seen from the following representation

$$R_k = (I - \gamma_k R_{k-1} e_k e_k^T) R_{k-1} = L_k R_{k-1}$$

with the $m \times m$ matrix L_k defined by

$$L_k = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & -\frac{(R_{k-1})_{k+1,k}}{(R_{k-1})_{kk}} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -\frac{(R_{k-1})_{mk}}{(R_{k-1})_{kk}} & & & 1 \end{bmatrix},$$

which differs from a Gauß matrix only in the position (k, k). It is known that during the *LU* decomposition the so-called *growth of entries* may happen. Note that this is reflected by the factor 2^k in (22). However, this growth is also known to be rarely observable in practice.

Numerical Example

In order to show that a thorough implementation of ACA can handle nonsmooth geometries, we consider the Dirichlet boundary value problem

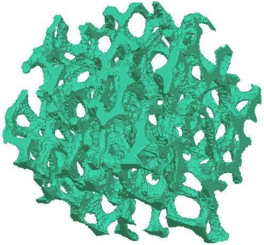
$$-\Delta u = 0 \quad \text{in } \Omega, \qquad u = g \quad \text{on } \Gamma.$$

Boundary integral equation for the unknown $t := \partial_\nu u$:

$$\mathcal{V}t = (\frac{1}{2}\mathcal{I} + \mathcal{K})g \quad \text{with} \quad \mathcal{V}u(y) = \frac{1}{4\pi}\int_\Gamma \frac{u(x)}{|x-y|}\,\mathrm{d}s_x$$

and $\mathcal{K}u(y) = \frac{1}{4\pi}\int_\Gamma u(x)\partial_{\nu_x}\frac{1}{|x-y|}\,\mathrm{d}s_x$. A Galerkin discretization with piecewise linears φ_j and piecewise constants ψ_i leads to

$$Vx = b, \quad b = (\frac{1}{2}M + K)\tilde{g},$$



where for $i = 1, \ldots, n'$ and $j = 1, \ldots, n$

$$V_{ij} = (\mathcal{V}\psi_j, \psi_i)_{L^2(\Gamma)}, \quad K_{ij} = (\mathcal{K}\varphi_j, \psi_i)_{L^2(\Gamma)}, \quad \text{and} \quad M_{ij} = (\varphi_j, \psi_i)_{L^2(\Gamma)}.$$

Furthermore, $\tilde{g} \in \mathbb{R}^n$ is the vector minimizing $\|g - \sum_{j=1}^n \tilde{g}_j\varphi_j\|_{L^2(\Gamma)}$. The solution x defines an approximation $t_h := \sum_{i=1}^{n'} x_i\psi_i$ of t.

Since $\mathcal{V}$ is coercive, its Galerkin stiffness matrix V is symmetric positive definite. Hence, in contrast to the $\mathcal{H}$-matrix approximant $K_{\mathcal{H}}$ to K, we may generate only the upper triangular part of the approximant $V_{\mathcal{H}}$ to V.

	$n = 28\,968$			$n = 115\,872$		
η	time	MB	ratio	time	MB	ratio
0.8	316s	259	8.1%	1 567s	1 264	2.5%
1.0	253s	204	6.4%	1 251s	995	1.9%
1.2	217s	173	5.4%	1 208s	967	1.9%
1.4	208s	162	5.1%	2 812s	2 513	4.9%

Table: Approximation results for V and $\varepsilon = 1_{10}-4$.

	$n = 28\,968$			$n = 115\,872$		
η	time	MB	ratio	time	MB	ratio
0.8	2 334s	543	17.4%	11 651s	2 789	5.5%
1.0	1 943s	443	14.2%	9 517s	2 264	4.4%
1.2	1 711s	386	12.3%	8 788s	2 119	4.2%
1.4	2 001s	475	15.2%	21 260s	6 222	12.2%

Table: Approximation results for K and $\varepsilon = 1_{10}-4$.

For the computation of the singular integrals we have used O. Steinbach's semi-analytic quadrature routines OSTBEM.

Helmholtz' equation

Boundary integral formulations can be derived and are particularly useful for Helmholtz' equation

$$\begin{aligned} -\Delta u - \omega^2 u &= 0 \quad \text{in } \Omega^c, \\ u &= 1 \quad \text{on } \partial\Omega, \end{aligned}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded domain and $\Omega^c = \mathbb{R}^d \setminus \overline{\Omega}$.
The mesh size h has to be chosen such that ωh is constant when discretizing $\partial\Omega$. As a consequence, $k \sim h^{-1} \sim n^{1/(d-1)}$ and methods based on low-rank approximants will not be able to achieve logarithmic-linear complexity for large ω. In the next table we compare the required rank for a prescribed accuracy $\varepsilon = 1_{10}{-}4$ of ACA applied to an admissible sub-block with the low-rank approximant resulting from the SVD for increasing wave numbers ω.

		SVD		ACA	
ω	matrix size	k	time	k	time
20	8 × 16	8	0.00s	8	0.00s
25	15 × 31	10	0.00s	13	0.00s
50	125 × 250	19	0.05s	26	0.00s
100	1000 × 2000	37	32.51s	42	0.05s
200	8000 × 16000	–	–	84	2.13s
400	64000 × 128000	–	–	175	100.82s

Using Hierarchical Matrices for Preconditioning

Consider a sequence of linear systems

$$A_n x_n = b_n, \quad n \to \infty, \tag{23}$$

where each $A_n \in \mathbb{C}^{n\times n}$ is invertible.

- convergence of Krylov subspace methods is determined by the distribution of eigenvalues of A_n;
- eigenvalues determined by mapping properties of the underlying differential or integral operator $\mathcal{A}$;
- a large condition number can arise even for small n.

Therefore, if (23) is to be solved iteratively, one has to incorporate a preconditioner.

IDEA:

- generate preconditioners C from low-accuracy approximations of the inverse of A: $C \approx A^{-1}$;
- Usually more efficient: approximate LU decomposition $C = (L_{\mathcal{H}} U_{\mathcal{H}})^{-1}$.

How much accuracy is required for preconditioning ?

Hermitian positive definite coefficient matrices

Use *preconditioned conjugate gradient method* (PCG) to solve Hermitian positive definite systems.

The condition

$$\|I - A_{\mathcal{H}}C\|_2 \leq \varepsilon < 1, \tag{24}$$

in which ε does not depend on n, leads to an well-conditioned matrix $A_{\mathcal{H}}C$.

The following lemma provides an estimate on the precision ε of the preconditioner.

Lemma

Assume that (24) *holds. Then*

$$\operatorname{cond}_2(A_{\mathcal{H}}C) = \|A_{\mathcal{H}}C\|_2 \|(A_{\mathcal{H}}C)^{-1}\|_2 \leq \frac{1+\varepsilon}{1-\varepsilon}.$$

Proof. The assertion follows from the triangle inequality

$$\|A_{\mathcal{H}}C\|_2 \leq \|I\|_2 + \|I - A_{\mathcal{H}}C\|_2 \leq 1 + \varepsilon$$

and from the Neumann series

$$\|(A_{\mathcal{H}}C)^{-1}\|_2 \leq \sum_{k=0}^{\infty} \|I - A_{\mathcal{H}}C\|_2^k = \frac{1}{1-\varepsilon}.$$

■

The choice $\varepsilon = 0.5$, for instance, guarantees that $\operatorname{cond}_2(A_{\mathcal{H}}C) \leq 3$.
→ problem independent convergence rates.
In order to be able to apply PCG, C additionally needs to be Hermitian positive definite. It is interesting to see that this is already guaranteed by condition (24).

Lemma

Assume that $A_{\mathcal{H}}$ is Hermitian positive definite. Then any Hermitian matrix C satisfying (24) *is positive definite, too.*

Proof. According to the assumptions, the square root $A_{\mathcal{H}}^{1/2}$ of $A_{\mathcal{H}}$ is defined. Since $A_{\mathcal{H}}C$ is similar to the Hermitian matrix $A_{\mathcal{H}}^{1/2}CA_{\mathcal{H}}^{1/2}$, the eigenvalues of $A_{\mathcal{H}}C$ are real. Moreover, for the smallest eigenvalue of $A_{\mathcal{H}}^{1/2}CA_{\mathcal{H}}^{1/2}$ it follows that

$$\lambda_{\min}(A_{\mathcal{H}}^{1/2}CA_{\mathcal{H}}^{1/2}) = \lambda_{\min}(A_{\mathcal{H}}C) \geq 1 - \varepsilon.$$

Let $x \neq 0$ and $y = A_{\mathcal{H}}^{-1/2}x$. Then $y \neq 0$ and we have

$$x^H Cx = y^H A_{\mathcal{H}}^{1/2}CA_{\mathcal{H}}^{1/2}y \geq (1-\varepsilon)\|y\|_2^2 > 0,$$

which proves that C is positive definite. ■

From the approximation by $\mathcal{H}$-matrices usually error estimates of the form

$$\|A_{\mathcal{H}} - C^{-1}\|_2 \leq \varepsilon \|A_{\mathcal{H}}\|_2 \quad \text{or} \quad \|A_{\mathcal{H}}^{-1} - C\|_2 \leq \varepsilon \|A_{\mathcal{H}}^{-1}\|_2 \tag{25}$$

instead of (24) are satisfied.

Lemma

Assume that (25) *holds with* $\varepsilon > 0$ *such that* $\varepsilon \operatorname{cond}_2(A_{\mathcal{H}}) < 1$. *Then*

$$\operatorname{cond}_2(A_{\mathcal{H}} C) \leq \frac{1 + \varepsilon \operatorname{cond}_2(A_{\mathcal{H}})}{1 - \varepsilon \operatorname{cond}_2(A_{\mathcal{H}})}.$$

Proof. Assume first that $\|A_{\mathcal{H}} - C^{-1}\|_2 \leq \varepsilon \|A_{\mathcal{H}}\|_2$. Since

$$\|I - (A_{\mathcal{H}} C)^{-1}\|_2 = \|(A_{\mathcal{H}} - C^{-1}) A_{\mathcal{H}}^{-1}\|_2 \leq \varepsilon \, \|A_{\mathcal{H}}\|_2 \|A_{\mathcal{H}}^{-1}\|_2 = \varepsilon \operatorname{cond}_2(A_{\mathcal{H}}),$$

one can apply the proof of the second last lemma with $A_{\mathcal{H}} C$ replaced by $(A_{\mathcal{H}} C)^{-1}$. If $\|A_{\mathcal{H}}^{-1} - C\|_2 \leq \varepsilon \|A_{\mathcal{H}}^{-1}\|_2$, then

$$\|I - A_{\mathcal{H}} C\|_2 = \|A_{\mathcal{H}}(A_{\mathcal{H}}^{-1} - C)\|_2 \leq \varepsilon \, \|A_{\mathcal{H}}\|_2 \|A_{\mathcal{H}}^{-1}\|_2 = \varepsilon \operatorname{cond}_2(A_{\mathcal{H}})$$

gives the assertion. ■

The stronger condition $\varepsilon \,\mathrm{cond}_2(A_{\mathcal{H}}) < 1$ implies that $\varepsilon \longrightarrow 0$ if $A_{\mathcal{H}}$ is not well-conditioned. This will NOT destroy the almost linear complexity since it will be seen that the complexity of the $\mathcal{H}$-matrix approximation depends logarithmically on the accuracy ε.

In the non-Hermitian case, not the spectral condition number of the coefficient matrix but the distance of a cluster of eigenvalues to the origin usually determines the convergence rate of appropriate Krylov subspace methods such as GMRes, BiCGStab, and MinRes. For the convergence of GMRes, for instance, the *numerical range*

$$F(A_{\mathcal{H}}C) := \left\{x^H A_{\mathcal{H}} C x : x \in \mathbb{C}^n,\ \|x\|_2 = 1\right\}$$

of $A_{\mathcal{H}}C$ is of particular importance. It is known that

$$\|b - A_{\mathcal{H}} x_k\|_2 \le 2 \left(\frac{r}{|z|}\right)^k \|b\|_2$$

provided $F(A_{\mathcal{H}}C) \subset B_r(z)$, where $B_r(z)$ denotes the closed disc around z with radius r. Condition (24) implies that $F(A_{\mathcal{H}}C) \subset B_\varepsilon(1)$, which follows from

$$|x^H A_{\mathcal{H}} C x - 1| = |x^H (A_{\mathcal{H}} C - I) x| \le \|I - A_{\mathcal{H}} C\|_2 \le \varepsilon \quad \text{for all } x \in \mathbb{C}^n,\ \|x\|_2 = 1.$$

Therefore, (24) also leads to a problem-independent convergence

$$\|b - A_{\mathcal{H}} x_k\|_2 \le 2\varepsilon^k \|b\|_2$$

of GMRes.

Industrial Applications

In the following numerical results we will use

$$C := U_{\mathcal{H}}^{-1} L_{\mathcal{H}}^{-1}$$

as an explicit preconditioner.

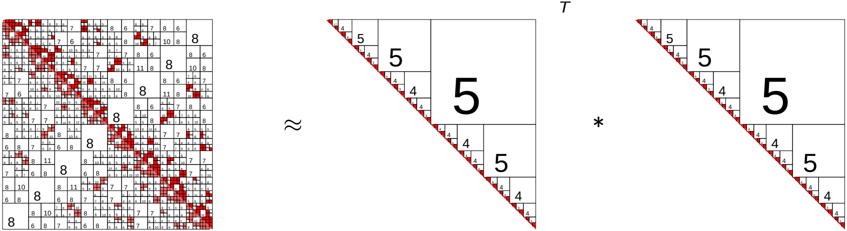


Figure: Low-precision Cholesky decomposition.

If $A_{\mathcal{H}}$ is Hermitian positive definite, $C := L_{\mathcal{H}}^{-T} L_{\mathcal{H}}^{-1}$ is used as a preconditioner.
The ability to compute preconditioners from the matrix approximant $A_{\mathcal{H}}$ in a black-box way is one of the advantages of $\mathcal{H}$-matrices over fast multipole methods.

The first example is the boundary integral equation

$$\frac{1}{2}u(y) + \int_\Gamma u(x)\frac{(\nu_x, y-x)}{|x-y|^3}\,\mathrm{d}s_x = \int_\Gamma \frac{\partial_{\nu_x}u(x)}{|x-y|}\,\mathrm{d}s_x, \quad y \in \Gamma,$$

with given Neumann boundary condition $\partial_\nu u = g$ on the surface Γ. The dimension of the coefficient matrix A arising from a collocation method is $n = 3760$. Since the kernel of A is one-dimensional (the surface is simply connected), the extended system

$$\begin{bmatrix} A & v \\ w^T & 0 \end{bmatrix}\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{26}$$

with $v \notin \operatorname{Im} A$ and $w \in \operatorname{Ker} A$, which is uniquely solvable, has to be considered instead. Note that (26) arises from adding the auxiliary condition $x \perp \operatorname{Ker} A$ by Lagrangian multipliers.

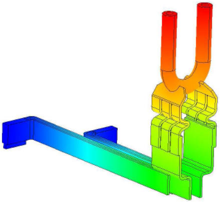


Figure: Device with electric potential (by courtesy of ABB Schweiz AG).

In the following table we compare the results obtained by fast methods and a standard solution strategy; i.e., A is built without approximation and the augmented system (26) is solved using Gaussian elimination. For the kernel vectors we have used $v = w = (1, \dots, 1)^T$.

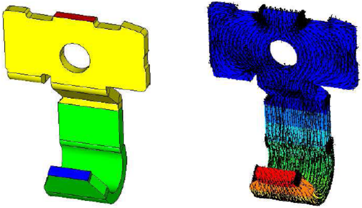
	storage		CPU time		
	matrix	precond.	matrix	precond.	solution
standard	108 MB		575.6s		1108.4s
Mbit	42 MB		149.1s		1273.2s
ACA	26 MB	12 MB	55.7s	1.7s	1.3s

In the row “Mbit”, the results using an implementation of the fast multipole method can be found.

The preconditioner was computed from the approximant $A_{\mathcal{H}}$ of A with precision $\delta = 0.1$. Although the double-layer potential operator is asymptotically well conditioned, the augmented coefficient matrix is ill-conditioned even for small problem sizes. This is due to the geometry and its discretization with strongly non-uniform triangles.

Mixed boundary value problems

We consider the following device, which is connected to an electric source of opposite voltages on the dark parts of the left picture.



The discrete single-layer potential operator V and the discrete hypersingular operator D are symmetric positive definite matrices. Hence, the coefficient matrix

$$A := \begin{bmatrix} -V & K \\ K^T & D \end{bmatrix}$$

is symmetric and non-singular since the Schur complement $S := D + K^T V^{-1} K$ of $-V$ in A is symmetric positive definite.

We employ the preconditioner

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

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

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

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

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

Theorem

The eigenvalues of $C^T A C$ are contained in

$$[-1 - c\delta, -1 + c\delta] \cup [\gamma_3^{-1}(1 - c\delta), 1 + c\delta],$$

where $c := 4(c_{\mathcal{K}} + 2) \max\{\|V^{-1}\|_2, \|D^{-1}\|_2\} \max\{1, \delta(c_{\mathcal{K}} + 2)\|V^{-1}\|_2\} + 1.$

Since $C^T AC$ is symmetric indefinite, we employ MinRes for the iterative solution of the preconditioned linear system. For the kth residual vector r_k it holds that

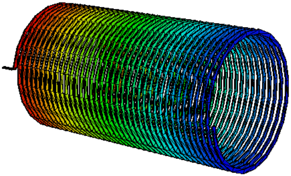
$$\|r_k\| \leq 2 \left(\frac{\sqrt{(a-\rho)(b+\rho)} - \sqrt{(a+\rho)(b-\rho)}}{\sqrt{(a-\rho)(b+\rho)} + \sqrt{(a+\rho)(b-\rho)}} \right)^{k/2} \|r_0\|, \quad k = 1, 2, \dots,$$

where the spectrum is assume to be enclosed in positive and negative intervals $[-a-\rho, -a+\rho] \cup [b-\rho, b+\rho]$. In order to obtain a convergence rate which is independent of the discretization, we therefore have to guarantee that $c\delta < \frac{1}{2}$.

$n = 5\,154$					$n = 20\,735$				
	precond.		solution			precond.		solution	
δ	time	MB	#It	time	δ	time	MB	#It	time
$5_{10}{-}2$	1.9s	6.7	11	0.5s	$5_{10}{-}3$	22.5s	35.6	9	3.0s
$1_{10}{-}1$	1.5s	5.6	17	0.6s	$1_{10}{-}2$	17.7s	34.7	13	4.1s
$2_{10}{-}1$	1.1s	4.5	23	0.8s	$5_{10}{-}2$	13.3s	30.5	24	5.9s

Mixed BVPs with vanishing Dirichlet part

The coarsest discretization of the following surface contains only 4 Dirichlet triangles. If the Dirichlet part would vanish completely, then the hypersingular operator would not be coercive at all.



After generating the approximant with accuracy $\varepsilon = 1_{10}{-}6$, we recompress a copy of the coefficient matrix to a blockwise relative accuracy δ. The hierarchical Cholesky decomposition fails to compute unless we use a stabilized variant which is based on the stabilization technique.

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

Iterating without any preconditioner does not converge at all.

Application to Finite Element Methods

In part we will apply $\mathcal{H}$-matrices to the finite element discretization of elliptic boundary value problems

$$\begin{aligned} \mathcal{L}u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

with bounded Lipschitz domains $\Omega \subset \mathbb{R}^d$, where $\mathcal{L}$ is a general uniformly elliptic second order partial differential operator in divergence form

$$\mathcal{L}u = -\text{div}[C\nabla u + \gamma u] + \beta \cdot \nabla u + \delta u$$

with coefficients $c_{ij}, \beta_i, \gamma_i, \delta \in L^\infty(\Omega)$, $i, j = 1, \dots, d$. The ellipticity of $\mathcal{L}$ is expressed by the assumption that for almost all $x \in \Omega$

$$0 < \lambda_{\mathcal{L}} \leq \lambda(x) \leq \Lambda_{\mathcal{L}}$$

for all eigenvalues $\lambda(x)$ of the symmetric matrix $C(x) \in \mathbb{R}^{d\times d}$ with entries c_{ij}.

Theorem

Assume that any Schur complement $S(b)$, $b \in P$ admissible, of a matrix A can be approximated by a matrix of rank k with accuracy ε such that $k \sim (\log n)^\alpha |\log \varepsilon|^\beta$, $\alpha, \beta > 0$. Then there are lower and upper triangular matrices $L_{\mathcal{H}}, U_{\mathcal{H}} \in \mathcal{H}(T_{I\times I}, k')$ with

$$k' \sim (\log n)^\alpha \left[|\log \varepsilon| + (\log n)^2 + (\log n)(\log \rho_n \text{cond}_2(A)) \right]^\beta$$

such that $\|A - L_{\mathcal{H}} U_{\mathcal{H}}\|_2 \leq \varepsilon$.

Comparison with multigrid methods

In the following example we demonstrate that varying coefficients really have an impact on the convergence properties of multigrid. We investigate the Dirichlet boundary value problem

$$\begin{aligned} -\operatorname{div} \alpha(x) \nabla u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

on the unit square $\Omega = (0,1) \times (0,1)$, where the coefficient α is defined by

$$\alpha(x) = \begin{cases} a, & x \in (\frac{1}{8}, \frac{1}{4}) \times (\frac{1}{8}, \frac{1}{4}), \\ 1, & \text{else.} \end{cases}$$

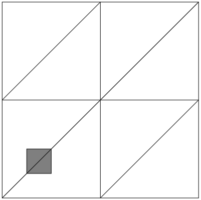


Figure: the coefficient α and the coarsest grid $\mathcal{T}_0$.

We use piecewise linear ansatz functions and apply the geometric multigrid procedure to a hierarchy of nested grids $\mathcal{T}_\ell$, $\ell = 4, 5, 6, 7$.

The convergence rates of the V- and the W-cycle using two Gauss-Seidel steps for pre- and postsmoothing, respectively, are shown in the following table.

	q (V-cycle)				q (W-cycle)				q ($\mathcal{H}$PCG)	
a	4	5	6	7	4	5	6	7	6	7
10^0	0.110	0.122	0.129	0.131	0.089	0.092	0.087	0.078	0.45	0.49
10^2	0.798	0.802	0.804	0.805	0.664	0.464	0.234	0.098	0.44	0.49
10^4	0.998	0.998	0.998	0.998	0.996	0.991	0.983	0.967	0.45	0.49
10^6	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.43	0.50

Table: Convergence rates q of multigrid and $\mathcal{H}$PCG.

For large a, the convergence of both the V- and the W-cycle slows down significantly, while $\mathcal{H}$PCG still gives reasonable convergence rates.

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

M. Bebendorf and W. Hackbusch.
Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients.
Numer. Math., 95(1):1–28, 2003.

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

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

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

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

W. Hackbusch and Z. P. Nowak.

On the fast matrix multiplication in the boundary element method by panel clustering.
Numer. Math., 54(4):463–491, 1989.

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

E. E. Tyrtyshnikov.
Mosaic-skeleton approximations.
Calcolo, 33(1-2):47–57 (1998), 1996.
Toeplitz matrices: structures, algorithms and applications (Cortona, 1996).