$\mathcal{H}^2$-matrices with adaptive bases

Steffen Börm

MPI für Mathematik in den Naturwissenschaften
Inselstraße 22–26, 04103 Leipzig
`http://www.mis.mpg.de/`

Goal: Treat certain **large** dense matrices in **linear** complexity.

Model problem: Discretization of integral operators of the form

$$\mathcal{K}[u](x) = \int_\Omega \kappa(x,y)u(y)\,dy.$$

Discretization: Galerkin method with FE basis $(\varphi_i)_{i=1}^n$ leads to a matrix $K \in \mathbb{R}^{n\times n}$ with

$$K_{ij} = \langle \varphi_i, \mathcal{K}[\varphi_j]\rangle_{L^2} = \int_\Omega \int_\Omega \varphi_i(x)\kappa(x,y)\varphi_j(y)\,dy\,dx. \quad (1 \le i,j \le n)$$

Problem: K is dense.

Idea: Use properties of the kernel function to approximate K.

Examples: Wavelets, multipole expansion, interpolation.

Idea: We consider a subdomain $\tau \times \sigma \subseteq \Omega \times \Omega$ and replace the kernel function κ by an interpolant

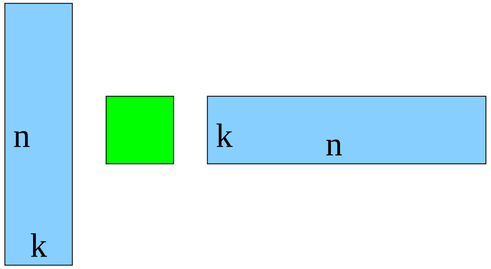
$$\tilde{\kappa}^{\tau,\sigma}(x,y) := \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \kappa(x_\nu^\tau, x_\mu^\sigma) \mathcal{L}_\nu^\tau(x) \mathcal{L}_\mu^\sigma(y). \qquad (k \ll n)$$

Result: Local matrix in factorized form

$$K_{ij}^{\tau,\sigma} := \int_\tau \int_\sigma \varphi_i(x)\kappa(x,y)\varphi_j(y)\,dy\,dx \approx \int_\tau \int_\sigma \varphi_i(x)\tilde{\kappa}^{\tau,\sigma}(x,y)\varphi_j(y)\,dy\,dx$$

$$= \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \underbrace{\kappa(x_\nu^\tau, x_\mu^\sigma)}_{=:S_{\nu\mu}^{\tau,\sigma}} \underbrace{\int_\tau \varphi_i(x)\mathcal{L}_\nu^\tau(x)\,dx}_{=:V_{i\nu}^\tau} \underbrace{\int_\sigma \varphi_j(y)\mathcal{L}_\mu^\sigma(y)\,dy}_{=:V_{j\mu}^\sigma}$$

$$= (V^\tau S^{\tau,\sigma} (V^\sigma)^\top)_{ij}$$



Problem: Typical kernel functions are **not globally smooth**.

Idea: They are **locally smooth** outside of the diagonal $x = y$. Examples: $1/\|x - y\|$, $\log \|x - y\|$.

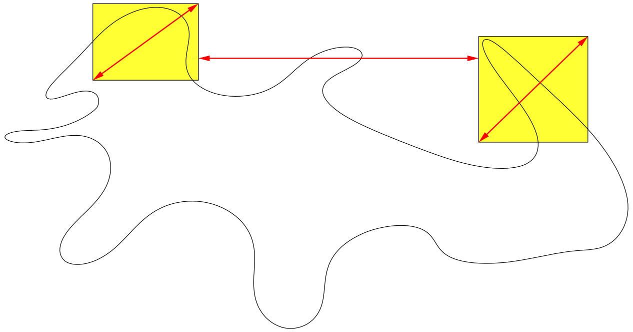
Error bound: For m-th order tensor interpolation on axis-parallel boxes $B^\tau \supseteq \tau$ and $B^\sigma \supseteq \sigma$, we get

$$\|\tilde{\kappa}^{\tau,\sigma} - \kappa\|_{L^\infty(B^\tau \times B^\sigma)} \leq C_{\mathrm{apx}}(m) \left(1 + \frac{2\,\mathrm{dist}(B^\tau, B^\sigma)}{\mathrm{diam}(B^\tau \times B^\sigma)}\right)^{-m}.$$

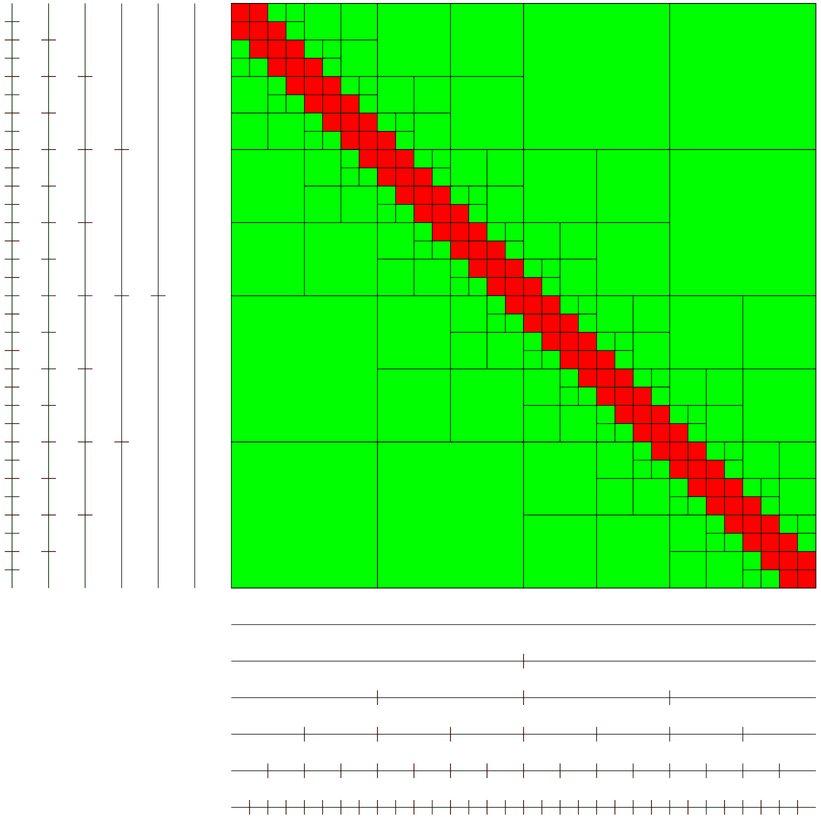
Rank: m interpolation points per coordinate $\Rightarrow k = m^d$.

Admissibility: Uniform exponential convergence requires

$$\mathrm{diam}(B^\tau \times B^\sigma) \leq 2\eta\,\mathrm{dist}(B^\tau, B^\sigma).$$



Approach: Split $\Omega \times \Omega$ into admissible subdomains and a small remainder.



Cluster tree: Hierarchy of subdomains of Ω

Nearfield: Small blocks, stored in standard format.

Farfield: Admissible blocks, stored in $V^\tau S^{\tau,\sigma}(V^\sigma)^\top$ format.

Partition: Collection of far- and nearfield blocks.

Construction of cluster tree and block partition can be accomplished by general algorithms in $\mathcal{O}(n)$ operations.

Goal: Treat the **cluster basis** $(V^\tau)_{\tau\in\mathcal{T}}$ efficiently.

Idea: For a cluster τ and $\tau' \in \operatorname{sons}(\tau)$ we have

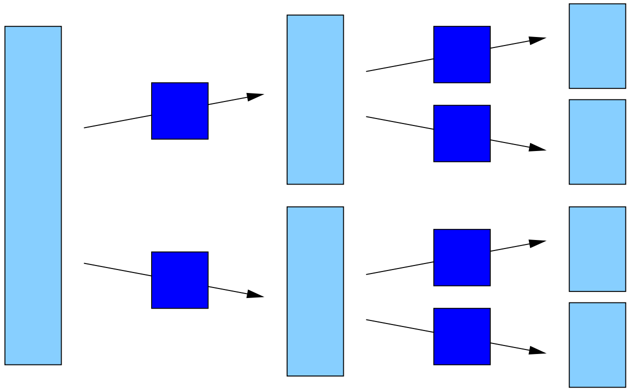
$$\mathcal{L}^\tau_\nu = \sum_{\nu'=1}^{k} \mathcal{L}^\tau_\nu(x^{\tau'}_{\nu'})\mathcal{L}^{\tau'}_{\nu'}. \qquad (1 \le \nu \le k)$$

Nested bases: Consider $i \in \tau' \subseteq \tau$. Setting $T^{\tau'}_{\nu'\nu} := \mathcal{L}^\tau_\nu(x^{\tau'}_{\nu'})$, we find

$$V^\tau_{i\nu} = \int_\tau \varphi_i(x)\mathcal{L}^\tau_\nu(x)\,dx = \sum_{\nu'=1}^{k} T^{\tau'}_{\nu'\nu}\int_\tau \varphi_i(x)\mathcal{L}^{\tau'}_{\nu'}(x)\,dx = (V^{\tau'}T^{\tau'})_{i\nu}.$$

Consequence: Store V^τ only for leaves and use $T^{\tau'}$ for all other clusters.

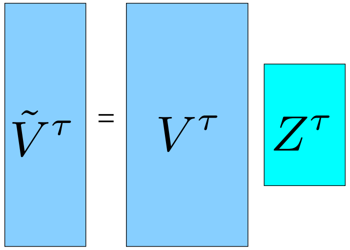
Complexity: Storage $\mathcal{O}(nk)$, matrix-vector $\mathcal{O}(nk)$.



Problem: Efficiency depends on the rank k. Can we reduce this rank by eliminating redundant expansion functions?

Idea: Orthogonalize columns of V^τ.

Gram-Schmidt: Find $\tilde V^\tau = V^\tau Z^\tau$ such that $(\tilde V^\tau)^\top \tilde V^\tau = I$ holds.



Nested structure: Using P^τ with $V^\tau = \tilde V^\tau P^\tau$, we find

$$V^\tau = \begin{pmatrix} V^{\tau_1} T^{\tau_1} \\ V^{\tau_2} T^{\tau_2} \end{pmatrix} = \begin{pmatrix} \tilde V^{\tau_1} P^{\tau_1} T^{\tau_1} \\ \tilde V^{\tau_2} P^{\tau_2} T^{\tau_2} \end{pmatrix} = \begin{pmatrix} \tilde V^{\tau_1} & \\ & \tilde V^{\tau_2} \end{pmatrix} \begin{pmatrix} P^{\tau_1} T^{\tau_1} \\ P^{\tau_2} T^{\tau_2} \end{pmatrix},$$

so it is sufficient to orthogonalize a $(2k) \times k$ matrix. The resulting cluster basis is nested due to

$$\tilde V^\tau = V^\tau Z^\tau = \begin{pmatrix} \tilde V^{\tau_1} (P^{\tau_1} T^{\tau_1} Z^\tau) \\ \tilde V^{\tau_2} (P^{\tau_2} T^{\tau_2} Z^\tau) \end{pmatrix} = \begin{pmatrix} \tilde V^{\tau_1} \tilde T^{\tau_1} \\ \tilde V^{\tau_2} \tilde T^{\tau_2} \end{pmatrix}.$$

Result: Complexity $\mathcal{O}(nk^2)$.

Example: Unit sphere, piecewise constant trial and test functions.

	Interpolation			Orthogonalized			
n	Build	MVM	Mem/n	Expl	Impl	MVM	Mem/n
2048	6	0.14	49.2	3	8	0.07	17.0
8192	26	0.54	45.6	15	35	0.31	21.1
32768	126	2.45	49.6	107	149	1.37	22.0
131072	765	10.33	53.4	469	774	5.48	22.0
524288	3347	43.08	53.8	2045	3473	23.81	22.0

$\mathcal{H}^2$-matrix constructed by cubic interpolation and subsequent orthogonalization, relative error $\leq 10^{-3}$.

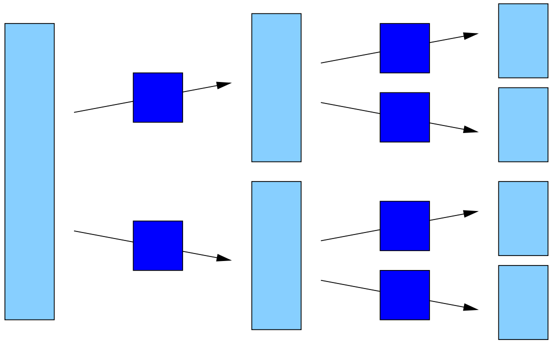
Hardware: SunFire 6800 with 900 MHz UltraSparc IIIcu processors

Software: HLib, see `http://www.hlib.org`

Problem: Interpolation does not take into account that a **subset** of polynomials may be sufficient to approximate the kernel function.

Idea: Use local singular value decompositions to find **optimal** cluster bases ⇒ Take kernel and geometry into account.

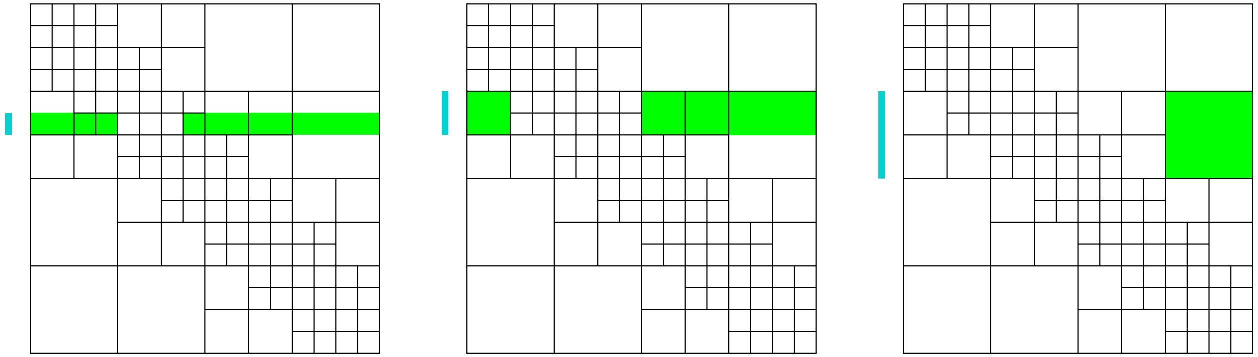
Important: Since the cluster bases are nested, the father clusters depend on the son clusters.



Consequence: If $\tau \times \sigma$ is a farfield block and $\tau' \in \operatorname{sons}(\tau)$, we have

$$\tilde{K}|_{\tau' \times \sigma} = (V^{\tau} S^{\tau,\sigma} (V^{\sigma})^{\top})|_{\tau' \times \sigma} = V^{\tau'} T^{\tau'} S^{\tau,\sigma} (V^{\sigma})^{\top},$$

i.e., the choice of $V^{\tau'}$ influences all farfield blocks connected to τ or its other ancestors.



Block row: $R^\tau := \{\sigma \; : \; \exists \tau^+ \supseteq \tau \; : \; \tau^+ \times \sigma \text{ is admissible}\}$.

Goal: Find an **orthogonal** matrix $V^\tau \in \mathbb{R}^{\tau \times k^\tau}$ that maximizes

$$\sum_{\sigma \in R^\tau} \|(V^\tau)^\top K|_{\tau\times\sigma}\|_F^2.$$

Optimal solution: Use eigenvectors of corresponding Gram matrix.

Nested structure: Ensured by bottom-up recursion and projections.

Result: Algorithm with complexity $\mathcal{O}(nk^2)$, truncation error can be controlled directly.

Example: Unit sphere, piecewise constant trial and test functions.

	Interpolation			Recompressed		
n	Build	MVM	Mem/n	Build	MVM	Mem/n
2048	6	0.14	49.2	11	0.01	3.7
8192	26	0.54	45.6	51	0.11	4.4
32768	128	2.41	49.6	210	0.45	4.6
131072	762	10.34	53.4	883	2.07	4.7
524288	3371	43.37	53.8	3818	9.77	5.2

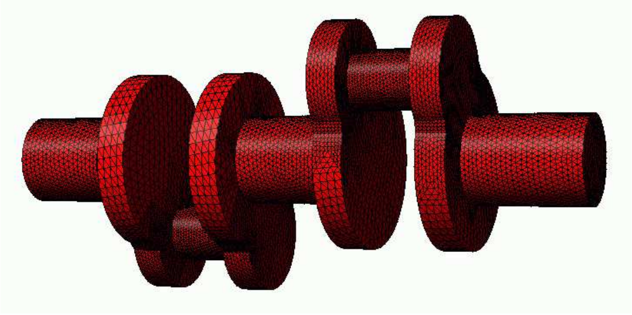
$\mathcal{H}^2$-matrix constructed by cubic interpolation.

Recompression with relative error bound 10^{-3}, maximal rank 16.

Hardware: SunFire 6800 with 900 MHz UltraSparc IIIcu processors.

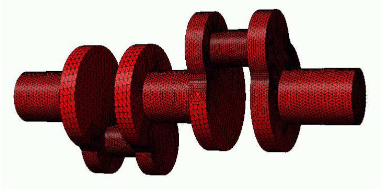
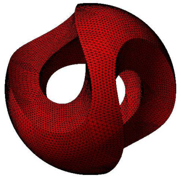
Software: HLib, see `http://www.hlib.org`

Example: Grid “Crank shaft” of the Netgen package.



n	1768	10126	113152
Build	20	200	2255
Mem/n	5.7	7.6	5.6
Near/n	4.2	4.7	3.4
MVM	0.02	0.24	2.18
Rel. error	0.05%	0.07%	0.08%
DMem/n	13.8	79.1	884.0

$$b(v,\phi) = \int_\Gamma\int_\Gamma \langle \nabla \times \phi(y), v(x)\rangle \langle \nabla_x \Phi(x,y), n(x)\rangle \, dy \, dx \\ - \int_\Gamma\int_\Gamma \langle \nabla \times \phi(y), n(x)\rangle \langle \nabla_x \Phi(x,y), v(x)\rangle \, dy \, dx$$

			
n	42432	169728	37632
Direct	1554	5178	1428
MVM	8.3	32.5	8.0
Mem/DoF	168	165	183
Recompressed	1700	6090	1644
MVM	0.8	2.6	0.7
Mem/DoF	8.0	6.5	8.0
Rel. error	8×10^{-3}		1×10^{-3}

$\mathcal{H}^2$-matrices: Nested structure leads to complexity $\mathcal{O}(nk)$ for storage requirements and matrix-vector multiplication.

Black box: The approximation is constructed by using only **kernel evaluations** $\kappa(x_\nu^\tau, x_\mu^\sigma)$ and works for all asymptotically smooth kernel functions.

Recompression: Use $\mathcal{O}(nk^2)$ algorithm to reduce the rank k, find a **quasi-optimal** cluster basis adapted to geometry, discretization and kernel function.

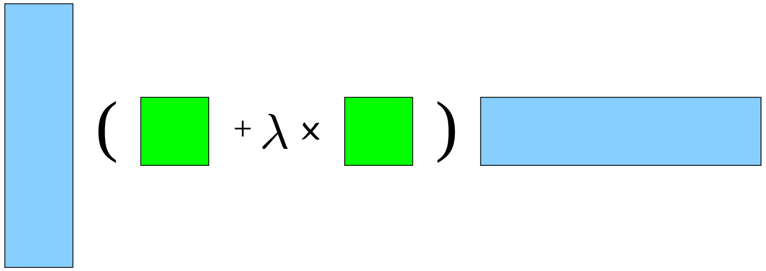
Work in progress: $\mathcal{H}^2$-matrix arithmetics, HCA.

Goal: Perform matrix addition and multiplication efficiently.

Addition and scaling: For an admissible block $\tau \times \sigma$, we have

$$(A + \lambda B)|_{\tau\times\sigma} = V^\tau (S_A^{\tau,\sigma} + \lambda S_B^{\tau,\sigma})(V^\sigma)^\top,$$

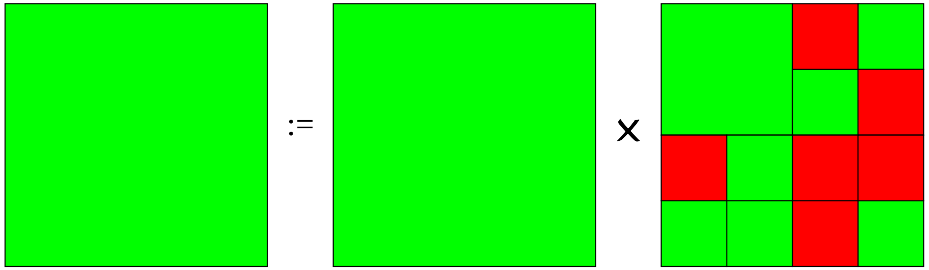
so $\mathcal{H}^2$-matrices form a vector space. Addition and scaling can be performed in $\mathcal{O}(nk)$ operations.



Multiplication: Defined recursively.

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

Use **truncation** if product does not match prescribed format.



Situation: Let $A = V^\tau S_A^{\tau,\varrho} (V^\varrho)^\top$, B not admissible.
Compute best approximation of AB in the form $C = V^\tau S_C^{\tau,\sigma} (V^\sigma)^\top$.

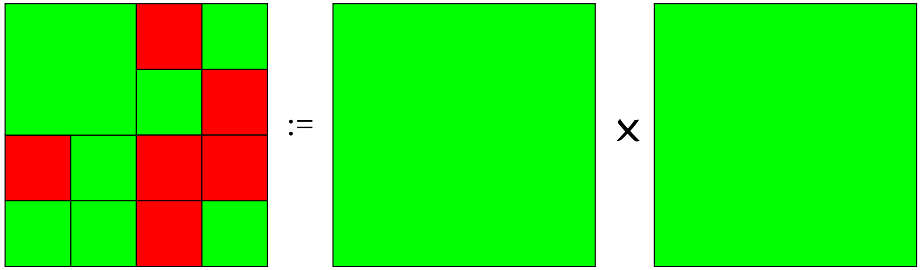
Orthogonality: If V^τ and V^σ are orthogonal:

$$S_C^{\tau,\sigma} = (V^\tau)^\top AB V^\sigma = (V^\tau)^\top V^\tau S_A^{\tau,\varrho} (V^\varrho)^\top B V^\sigma = S_A^{\tau,\varrho} \underbrace{(V^\varrho)^\top B V^\sigma}_{=:\tilde{S}_B^{\varrho,\sigma}}.$$

Problem: Computing $(V^\varrho)^\top B V^\sigma$ directly is too expensive.

Idea: Prepare $\tilde{S}_B^{\varrho,\sigma} := (V^\varrho)^\top B V^\sigma$ in advance.

Matrix forward transformation: Since the cluster bases are nested, all matrices $\tilde{S}_B^{\varrho,\sigma}$ can be prepared in $\mathcal{O}(nk^3)$ operations.



Situation: Let $A = V^\tau S_A^{\tau,\varrho}(V^\varrho)^\top$ and $B = V^\varrho S_B^{\varrho,\sigma}(V^\sigma)^\top$.

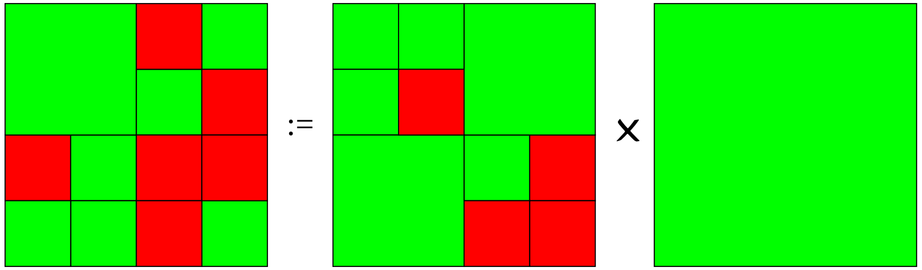
Advantage: The product AB already has the factorized form

$$AB = V^\tau S_A^{\tau,\varrho}(V^\varrho)^\top V^\varrho S_B^{\varrho,\sigma}(V^\sigma)^\top = V^\tau (S_A^{\tau,\varrho} S_B^{\varrho,\sigma})(V^\sigma)^\top.$$

Problem: Converting AB directly to the format of C will lead to inacceptable complexity.

Idea: Store result in $\tilde{S}_C^{\tau,\sigma} := S_A^{\tau,\varrho} S_B^{\varrho,\sigma}$ and distribute after the multiplication is complete.

Matrix backward transformation: Since the cluster bases are nested, all matrices $\tilde{S}_C^{\tau,\sigma}$ can be distributed in $\mathcal{O}(nk^3)$ operations.



Situation: Let $B = V^{\varrho} S_B^{\varrho,\sigma} (V^{\sigma})^{\top}$.

Approach: Split B and proceed by recursion:

$$B|_{\varrho'\times\sigma'} = V^{\varrho'} \underbrace{T^{\varrho'} S_B^{\varrho,\sigma} (T^{\sigma'})^{\top}}_{=:\widehat{S}_B^{\varrho',\sigma'}} (V^{\sigma'})^{\top}.$$

Complexity: Only $\mathcal{O}(C_{\mathrm{sp}}^2 n)$ such splitting operations are required $\Rightarrow$ Complexity $\mathcal{O}(C_{\mathrm{sp}}^2 n k^3)$.

Result: The best approximation C of AB in a prescribed $\mathcal{H}^2$-matrix format can be computed in $\mathcal{O}(C_{\mathrm{sp}}^2 n k^3)$ operations.

Example: Discretized double layer potential on the unit sphere K, approximate KK.

n	Build	C_{sp}	$\mathcal{H}^2$-MVM	$\mathcal{H}^2$-MMM	$\mathcal{H}$-MMM	Error
512	1.7	16	< 0.01	0.3	0.7	1.1_{-2}
2048	10.7	36	0.01	6.0	39.0	1.2_{-2}
8192	49.4	42	0.10	35.6	315.9	4.5_{-3}
32768	206.3	56	0.45	172.3	2085.2	4.9_{-3}
131072	858.3	84	2.00	924.2	13213.8	8.6_{-5}

Important: Here, the result matrix is projected to the same cluster bases and the same block structure as K.

Question: Can we reduce the error by using adapted cluster bases?

Modification: Choose cluster bases for result matrix adaptively to improve the precision.

n	C_{sp}	Old MMM	Error	New basis	New MMM	Error
512	16	0.3	1.1_{-2}	0.3	0.3	5.0_{-5}
2048	36	6.0	1.2_{-2}	9.5	6.8	6.6_{-5}
8192	42	35.6	4.5_{-3}	98.9	40.8	1.4_{-4}
32768	56	172.3	4.9_{-3}	878.9	194.7	4.3_{-6}
131072	84	924.2	8.6_{-5}	5030.2	966.9	2.9_{-6}

Advantage: Precision significantly improved, computation time only slightly increased.

Current work: Improve the efficiency of the construction of adapted cluster bases.

Problem: For each admissble block $V^\tau S^{\tau,\sigma}(V^\sigma)^\top$ of the $\mathcal{H}^2$-matrix, the kernel function κ has to be evaluated in m^{2d} points.
⇒ Very time-consuming if high precision is required.

Approach: Apply adaptive cross approximation to the coefficient matrices $S^{\tau,\sigma}$ to find rank k representation $S^{\tau,\sigma} \approx X^{\tau,\sigma}(Y^{\tau,\sigma})^\top$.

Result: Only $m^d k \ll m^{2d}$ kernel evaluations required to approximate coefficient matrix $S^{\tau,\sigma}$.

Important: ACA is only applied to coefficient matrices.
⇒ Construction still linear in n, no problems with quadrature.

m	5	6	7	8
Std	$24.1s$	$64.7s$	$160.8s$	$399.2s$
HCA	$22.3s$	$49.2s$	$90.0s$	$222.7s$

$\mathcal{H}^2$-matrices: Data-sparse approximation of certain dense matrices with **linear** complexity in the number of degrees of freedom.

Adaptive cluster bases: Find quasi-optimal expansion systems for arbitrary matrices. In the case of integral operators, **point evaluations** of the kernel function are sufficient to construct efficient and reliable $\mathcal{H}^2$-matrix approximations.

$\mathcal{H}^2$-matrix arithmetics: Approximate $A + \lambda B$ and AB in $\mathcal{O}(nk^2)$ and $\mathcal{O}(nk^3)$ operations. Next goal: Approximate A^{-1} and LU.

HCA: Reduce the number of kernel evaluations required to construct the $\mathcal{H}^2$-matrix.

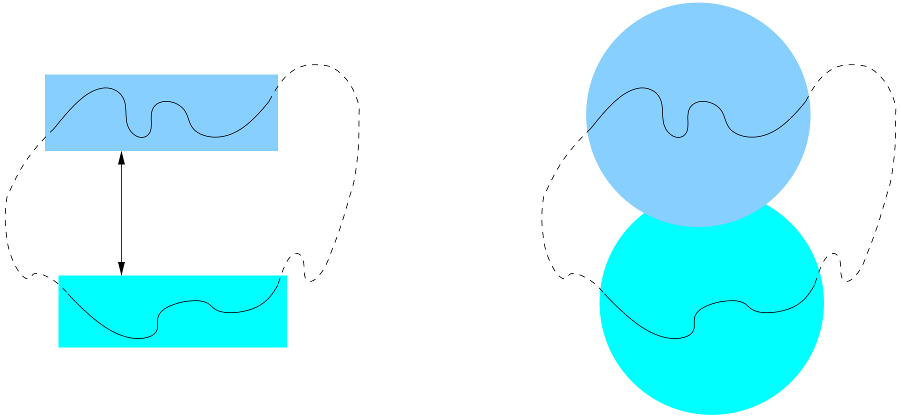
Software: HLib package for hierarchical matrices and $\mathcal{H}^2$-matrices, including applications to elliptic PDEs and integral equations.

`http://www.hlib.org`

Construction: Pointwise evaluation and integrals of polynomials.

$$T^{t'}_{\nu'\nu} = \mathcal{L}^t_\nu(x^{t'}_{\nu'}), \quad S^{t,s}_{\nu\mu} = \kappa(x^t_\nu, x^s_\mu), \quad V^t_{i\nu} = \int_\Gamma \varphi_i(x)\mathcal{L}^t_\nu(x)\,dx.$$

Interpolation	**Multipole**
– Rank $k = p^d$	+ Rank $k = p^{d-1}$
– Low frequencies only	+ Low and high frequencies
+ General kernels	– Special kernels only
+ Anisotropic clusters	– Spherical clusters only



Goal: Reduce rank for interpolation without sacrificing precision.