Söllerhaus Workshop '04: Adaptive Fast BEM in Industrial Applications

Hybrid Cross Approximation

joint work with Steffen Börm

Lars Grasedyck

Max Planck Institute for
Mathematics in the Sciences
Leipzig

1. Model problem: SLP / DLP
2. Data-sparse $\mathcal{H}$-matrix format
3. $\mathcal{H}$-matrix coarsening & arithmetic
4. Hybrid Cross Approximation

$$
\begin{aligned}
-\Delta u = 0 &\qquad \text{in unit sphere } \Omega := B(0,1) \subset \mathbb{R}^3, \\
u = u_D &\qquad \text{on } \Gamma := \partial\Omega \quad \textit{or} \\
\partial_n u = u_N &\qquad \text{on } \Gamma
\end{aligned}
$$

Solution u fulfils for $x \in \Gamma$

$$
\frac{1}{2}u(x) = \underbrace{\frac{1}{4\pi}\int_\Gamma \frac{u_N(y)}{\|x-y\|}\,\mathrm{d}\Gamma_y}_{=:\mathsf{V}[u_N]} - \underbrace{\frac{1}{4\pi}\int_\Gamma \frac{\langle n(y), x-y\rangle u_D(y)}{\|x-y\|^3}\,\mathrm{d}\Gamma_y}_{=:\mathsf{K}[u_D]}
$$

On the boundary Γ we get the two mappings

$$
\begin{aligned}
u_D &= \left(\frac{1}{2} + K\right)^{-1} V u_N, \\
u_N &= V^{-1}\left(\frac{1}{2} + K\right) u_D.
\end{aligned}
$$

Goal: discretise V and K efficiently / solve linear system

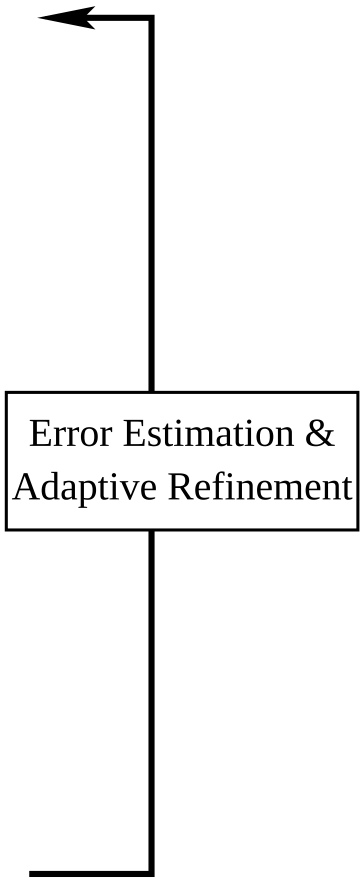
$$V[u_N] = \int_\Gamma g_V(x,y)u_N(y)\,\mathrm{d}\Gamma_y, \qquad K[u_D] = \int_\Gamma g_K(x,y)u_D(y)\,\mathrm{d}\Gamma_y$$

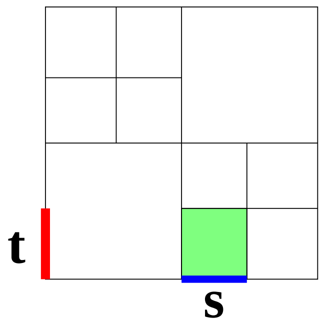
Discretisation:

- discretisation by Galerkin's method
  $\rightarrow$ dense stiffness matrix A
- compression by ACA/Interpolation/...
  $\rightarrow$ data-sparse $\mathcal{H}$-matrix $A_\mathcal{H}$
- algebraic recompression/coarsening
  $\rightarrow$ coarse $\mathcal{H}$-matrix $\tilde{A}_\mathcal{H}$

Solution:

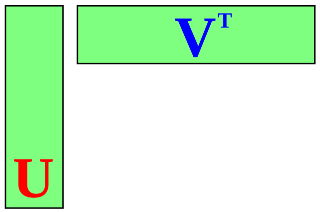
- (algebraic recompression/coarsening)
- approximate LU-dec. $\tilde{A}_\mathcal{H} \approx L_\mathcal{H} U_\mathcal{H}$
- $L_\mathcal{H} U_\mathcal{H}$-preconditioned GMRES





$$A_{ij} = \int_\Gamma \int_\Gamma \phi_i(x) g(x,y) \phi_j(y) \, \mathrm{d}\Gamma_x \mathrm{d}\Gamma_y$$

$$A|_{t\times s} \approx UV^T, \qquad U, V \in \mathbb{R}^{n\times k}.$$



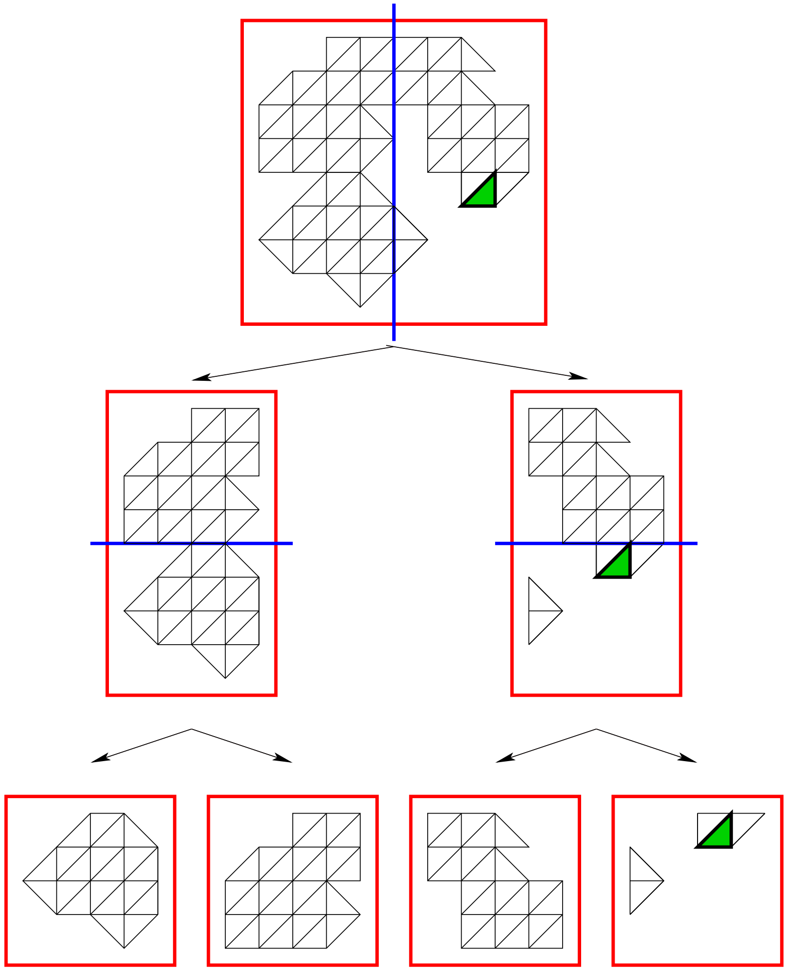
Interpolation:

$$g(x,y) \approx \sum_{\nu=1}^{m^3} L_\nu(x) g(x_\nu, y)$$

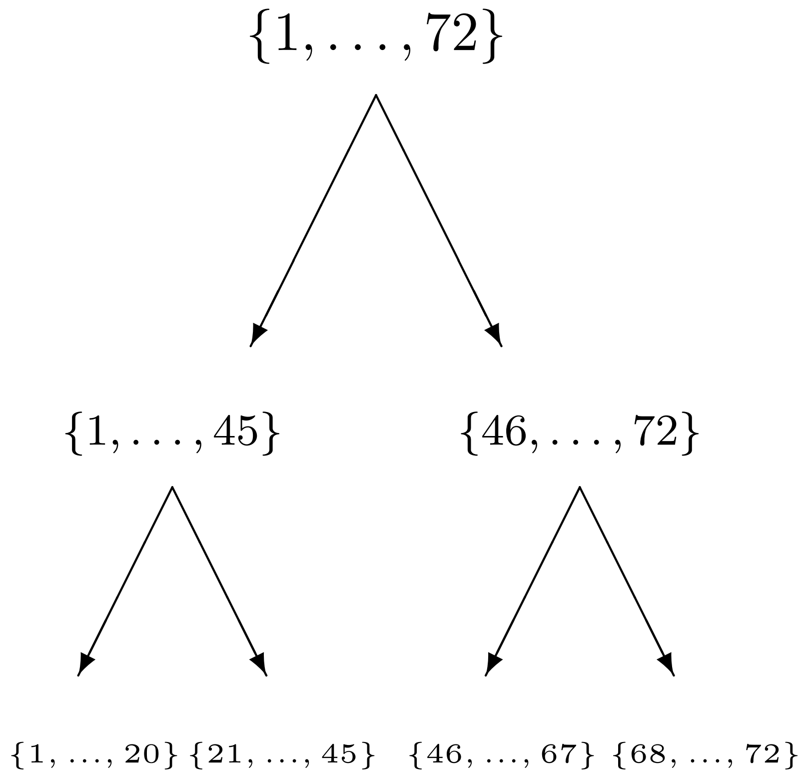
$$U_{i\nu} = \int_\Gamma \phi_i(x) L_\nu(x) \, \mathrm{d}\Gamma_x, \quad V_{j\nu} = \int_\Gamma \phi_j(y) g(x_\nu, y) \, \mathrm{d}\Gamma_y$$

- Exponential convergence requires admissibility condition
- Rigorous proof for asymptotically smooth kernels
- Works also for double layer

1. Subdivision of the index set $\mathcal{I} := \{1, \ldots, 72\}$



Geometry → BinarySpacePartitioning

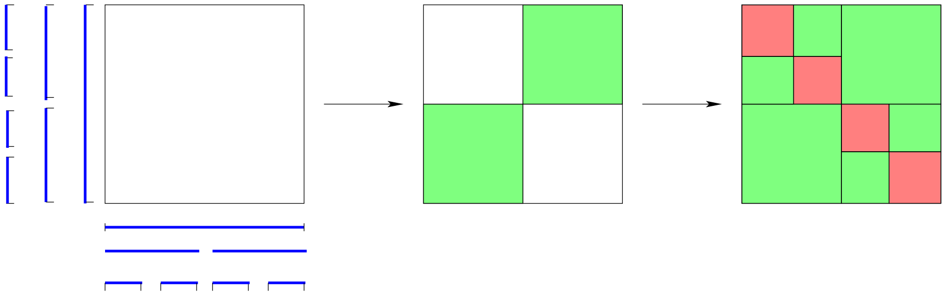


Indices → cluster tree $T_{\mathcal{I}}$

2. Subdivision of the product index set $\mathcal{I} \times \mathcal{I}$

Given: cluster tree $T_{\mathcal{I}}$ with root $\mathcal{I} = \{1, \ldots, n\}$

Seeking: block cluster tree $T_{\mathcal{I} \times \mathcal{I}}$

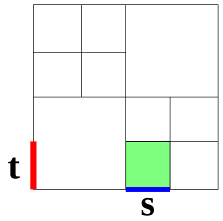


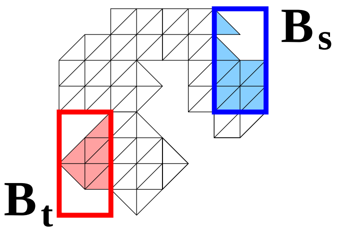
Start: $\mathcal{I} \times \mathcal{I}$. Iterate: subdivide inadmissible blocks:

$$\text{sons}(t \times s) := \text{sons}(t) \times \text{sons}(s).$$

η - Admissibility:

$\min(\text{diam}(B_t), \text{diam}(B_s)) \leq \eta \, \text{dist}(B_t, B_s)$

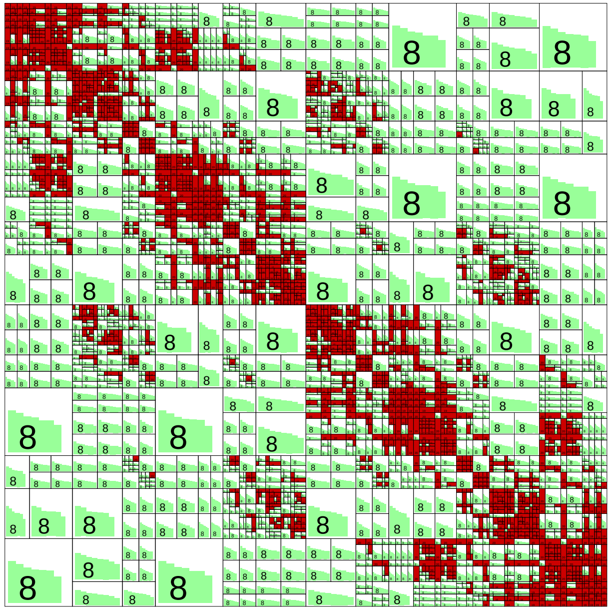




Model problem: DLP on the sphere

Integration: automatic quadrature [Erichsen/Sauter]

DLP, $n = 2048$



Initial $\mathcal{H}$-matrix

Interpol.	Build [Sec.]	Storage [KB/DoF]	Error $\|I - A_{\mathcal{H}}^{-1} A\|$
$n = 8K$	44	10.2	7.2×10^{-3}
$n = 32K$	241	29.7	6.1×10^{-3}
$n = 128K$	1353	40.9	5.7×10^{-3}
ACA			
$n = 8K$	12	5.9	9.1×10^{-4}
$n = 32K$	58	7.1	1.0×10^{-3}
$n = 128K$	284	8.3	2.5×10^{-3}

Complexity Interpolation: $\mathcal{O}\left(n \log(n) \log(\varepsilon)^3\right)$

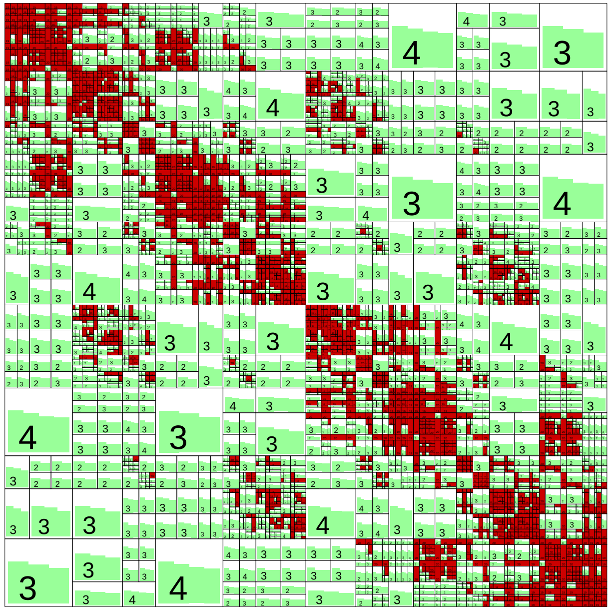
ACA: $\approx \mathcal{O}\left(n \log(n) \log(\varepsilon)^4\right)$

Idea for 1st Recompression (blockwise):

Given: UV^T → Compute SVD

→ discard small singular values $\sigma_i < \varepsilon\,\sigma_1$

DLP, $n = 2048$

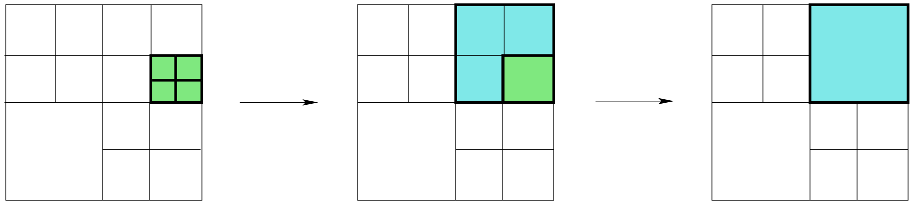


1. Recompression

Interpol.	1.Rec. [Sec.]	Storage [KB/DoF]	Error $\|I - A_{\mathcal{H}}^{-1}A\|$
$n = 8K$	9	4.2	7.1×10^{-3}
$n = 32K$	48	4.7	7.2×10^{-3}
$n = 128K$	262	5.5	5.5×10^{-3}
ACA			
$n = 8K$	1	4.3	3.8×10^{-3}
$n = 32K$	7	4.8	3.7×10^{-3}
$n = 128K$	30	5.4	3.9×10^{-3}

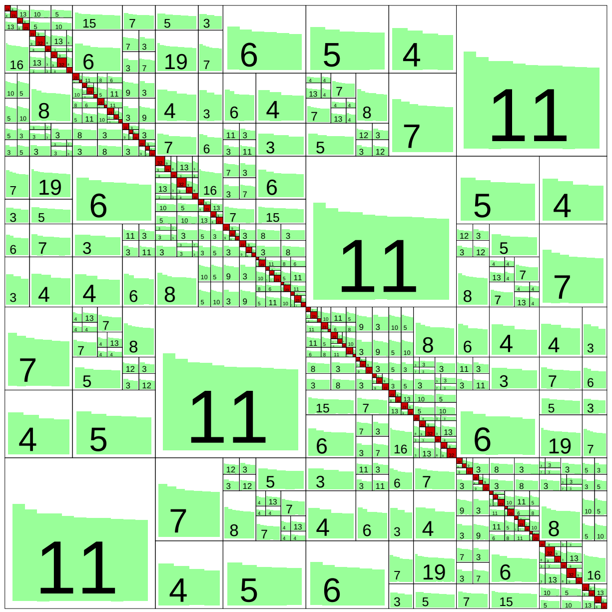
Complexity 1st Recompression: $\mathcal{O}\left(n\log(n)k^2\right)$

Idea for Coarsening (leaves to root):



Join 4 sons $\rightarrow$ SVD $\rightarrow$ discard singular values $\sigma_i < \tilde{\varepsilon}\,\sigma_1$

DLP, $n = 2048$



Coarsening

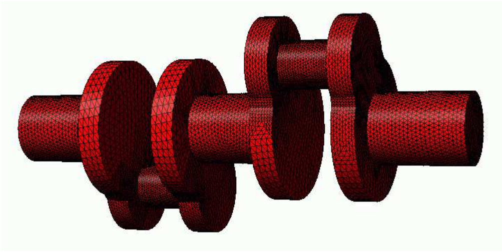
Interpol.	2.Rec [Sec.]	Storage [KB/DoF]	Error $\|I - A_{\mathcal{H}}^{-1} A\|$
$n = 8K$	12	1.9	8.0×10^{-3}
$n = 32K$	54	2.3	7.8×10^{-3}
$n = 128K$	224	3.0	5.7×10^{-3}
ACA			
$n = 8K$	13	1.8	6.0×10^{-3}
$n = 32K$	53	2.4	5.5×10^{-3}
$n = 128K$	223	2.9	5.4×10^{-3}

Complexity Coarsening: $\mathcal{O}\left(n \log(n) k^2\right)$

Example "Crank Shaft"

$n = 25744$

SLP, $\|I - VV_{\mathcal{H}}^{-1}\|_2 \approx 10^{-3}$



	Assembly	Coarsen	Cholesky	Solve	
standard	298	0	0	156	(81)
no prec.	298	86	0	46	(81)
$\varepsilon = 0.02$	298	86	31	6.7	(9)
$\varepsilon = 0.00001$	298	86	213	0.3	

Software: HLib [http://www.hlib.org] Hardware: SUNFIRE 900 MHz

First paper on $\mathcal{H}$-matrices: Hackbusch '99

2d/3d model problem: Hackbusch/Khoromskij '00

General Arithmetic/Complexity: Hackbusch, G. '00-'01

- successive rank 1 approximation of M
- each rank 1 term is of the form

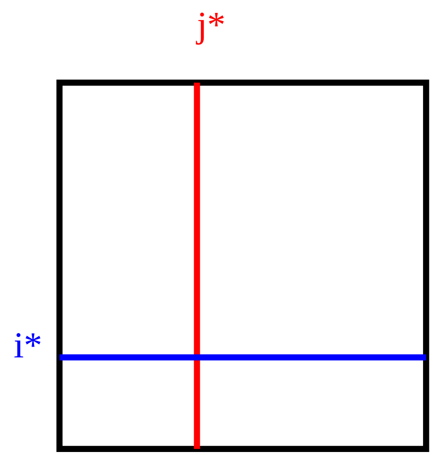
$$R^{\nu}_{i,j} = M_{i,j^*} M_{i^*,j} / M_{i^*,j^*}$$

with pivot index i^* determined by some heuristic and

$$j^* := \mathrm{argmax}_j M_{i^*,j}.$$

E.g., choice of i^* in next step:

$$i^* := \mathrm{argmax}_i M_{i,j^*}.$$



- only few entries $M_{i^*,j}, M_{i,j^*}$ needed
- result: rank k matrix $R = \sum R^{\nu}$

DLP/Galerkin		[Sec.]	[KB/DoF]	$\|I - (\tilde{G})^{-1} G\|_2$
Sphere	ACA, $\varepsilon = 10^{-2}$	392	15.9	2.2×10^{-3}
	ACA, $\varepsilon = 10^{-3}$	459	18.9	2.6×10^{-4}
	ACA, $\varepsilon = 10^{-4}$	548	22.9	3.2×10^{-5}
	ACA, $\varepsilon = 10^{-5}$	649	27.1	1.2×10^{-6}
Sphere	Interpol., $m = 1$	438	29.2	4.5×10^{-2}
	Interpol., $m = 2$	964	54.1	2.7×10^{-3}
	Interpol., $m = 3$	2231	61.5	1.7×10^{-4}
	Interpol., $m = 4$	3304	75.5	1.2×10^{-5}
Cube	ACA, $\varepsilon = 10^{-2}$	791	14.8	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-3}$	894	18.2	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-4}$	1034	22.6	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-5}$	1202	27.4	1.8×10^{-2}
Cube	Interpol., $m = 1$	527	28.7	5.3×10^{-2}
	Interpol., $m = 2$	1349	55.1	7.5×10^{-3}
	Interpol., $m = 3$	3059	71.7	1.6×10^{-3}
	Interpol., $m = 4$	5126	89.4	6.8×10^{-5}

What is proven [Bebendorf '00] ?

- if

$$M_{ij} = g(x_i, y_j)$$

  $x_i \in X$, $y_j \in Y$ for some asymptotically smooth g and

$$\min\{\mathrm{diam}(X), \mathrm{diam}(Y)\} \leq \eta \, \mathrm{dist}(X, Y)$$

  then the error $M_{ij} - R_{ij}$ after k ACA steps is at most

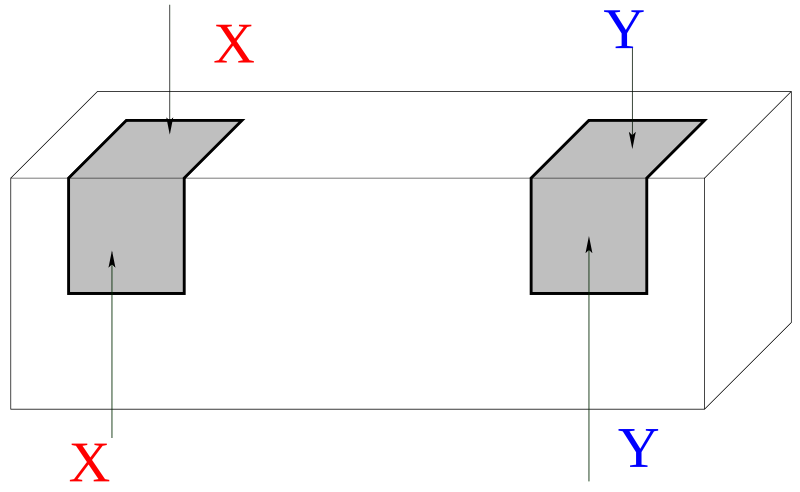
$$\varepsilon = \mathcal{O}(2^k \eta^{\sqrt[3]{k}})$$

- in practice growth factor 1, convergence $(1 + c\eta^{-1})^{-\sqrt{k}}$
- proof works only for Nystrøm of SLP
- no proof for DLP or Galerkin

Example (DLP):

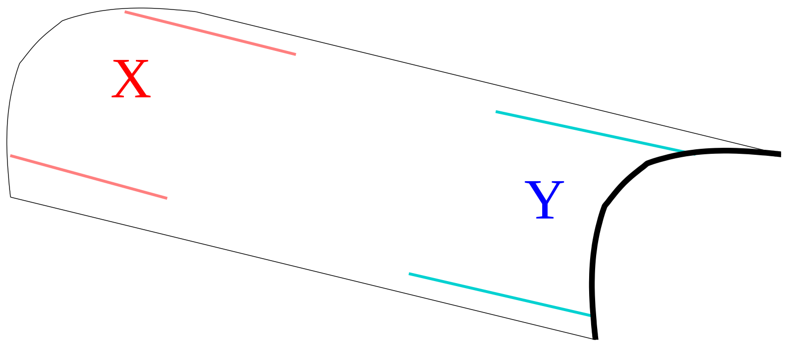
$$M_{i,j} = \frac{\langle n(y_j), x_i - y_j \rangle}{\|x_i - y_j\|^3}$$

$$M = \begin{bmatrix} M_{11} & 0 \\ 0 & M_{22} \end{bmatrix}$$



- ACA approximates either M_{11} or M_{22}
- Error $\|M - R\|_2 = \|M\|_2$.
- kernel function g is asymptotically smooth in x but not y
- there exists a low rank approximation
- ACA doesn't find it
- standard error estimator indicates success

$$M_{i,j} = \frac{\langle n(y_j), x_i - y_j \rangle}{\|x_i - y_j\|^3}$$



$$M = \begin{bmatrix} M_{11} & 0 \\ 0 & M_{22} \end{bmatrix}$$

- ACA approximates either M_{11} or M_{22}
- Error $\|M - R\|_2 = \|M\|_2$.
- kernel function g is asymptotically smooth in x but not y
- surface is smooth
- there exists a low rank approximation
- ACA doesn't find it
- standard error estimator indicates success

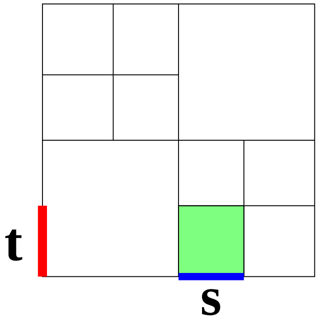
Conclusions:

- ACA is proven for $M_{ij} = g(x_i, y_j)$
- The heuristic works well for many problems
- In general it may fail
- No guaranteed error estimate

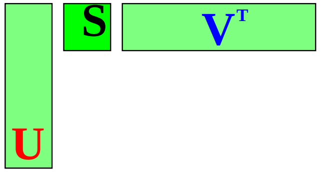
What we want:

- a method like ACA but
- which cannot fail $\Rightarrow$ proof
- works for SLP, DLP, Collocation, Galerkin
- is simple and easy to implement
- is as fast as the ACA heuristic

HCA(I):



$$A_{ij} = \int_\Gamma \int_\Gamma \phi_i(x) g(x,y) \phi_j(y) \; \mathrm{d}\Gamma_x \mathrm{d}\Gamma_y$$



$$A|_{t\times s} \approx USV^T, \qquad U, V \in \mathbb{R}^{n\times k}.$$

Complete interpolation:

$$g(x,y) \approx \sum_{\nu=1}^{m^3} \sum_{\mu=1}^{m^3} L_\nu(x) g(x_\nu, y_\mu) L_\mu(y)$$

$$U_{i\nu} = \int_\Gamma \phi_i(x) L_\nu(x) \; \mathrm{d}\Gamma_x, \quad V_{j\nu} = \int_\Gamma \phi_j(y) L_\mu(y) \; \mathrm{d}\Gamma_y$$

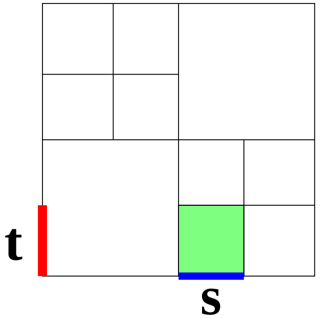
Approximate the coupling matrix by ACA:

$$S \approx AB^T, \qquad S_{\nu,\mu} = g(x_\nu, y_\mu)$$

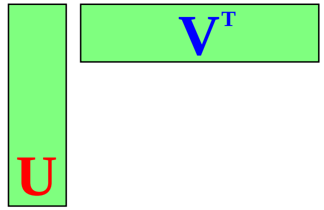
DLP: apply normal derivative to L_μ

DLP/Galerkin		[Sec.]	[KB/DoF]	$\|I-(\tilde{G})^{-1}G\|_2$
Sphere	ACA, $\varepsilon = 10^{-2}$	392	15.9	2.2×10^{-3}
	ACA, $\varepsilon = 10^{-3}$	459	18.9	2.6×10^{-4}
	ACA, $\varepsilon = 10^{-4}$	548	22.9	3.2×10^{-5}
	ACA, $\varepsilon = 10^{-5}$	649	27.1	1.2×10^{-6}
Sphere	HCA(I), $m = 1$	315	18.5	7.1×10^{-2}
	HCA(I), $m = 2$	411	23.0	4.2×10^{-3}
	HCA(I), $m = 3$	780	27.7	4.4×10^{-4}
	HCA(I), $m = 4$	1361	32.8	2.9×10^{-5}
Cube	ACA, $\varepsilon = 10^{-2}$	791	14.8	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-3}$	894	18.2	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-4}$	1034	22.6	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-5}$	1202	27.4	1.8×10^{-2}
Cube	HCA(I), $m = 1$	346	11.5	1.6×10^{-1}
	HCA(I), $m = 2$	444	17.1	3.7×10^{-2}
	HCA(I), $m = 3$	901	20.5	5.3×10^{-3}
	HCA(I), $m = 4$	1627	27.7	4.0×10^{-4}

HCA(II):



$$A_{ij} = \int_\Gamma \int_\Gamma \phi_i(x) g(x,y) \phi_j(y) \; \mathrm{d}\Gamma_x \mathrm{d}\Gamma_y$$



$$A|_{t\times s} \approx UV^T, \qquad U, V \in \mathbb{R}^{n\times k}.$$

Compute coefficients $C_{\ell,q}$, $D_{\ell,q}$ by applying ACA to the coupling matrix S to get

$$g(x,y) \approx \sum_{\ell=1}^{k} \left(\sum_{q=1}^{\ell} g(x, y_q) C_{\ell,q} \right) \left(\sum_{q=1}^{\ell} g(x_q, y) D_{\ell,q} \right)$$

Compute the matrices U and V with entries

$$U_{i\ell} := \sum_{q=1}^{\ell} \int_{\Omega_t} \phi_i(x) g(x, y_{\nu_q}) C_{\ell,q}, \qquad V_{j\ell} := \sum_{q=1}^{\ell} \int_{\Omega_s} \phi_j(y) g(x_{\mu_q}, y) D_{\ell,q}$$

DLP/Galerkin		[Sec.]	[KB/DoF]	$\|I-(\tilde{G})^{-1}G\|_2$
Sphere	ACA, $\varepsilon = 10^{-2}$	392	15.9	2.2×10^{-3}
	ACA, $\varepsilon = 10^{-3}$	459	18.9	2.6×10^{-4}
	ACA, $\varepsilon = 10^{-4}$	548	22.9	3.2×10^{-5}
	ACA, $\varepsilon = 10^{-5}$	649	27.1	1.2×10^{-6}
Sphere	HCA(II), $m = 1$	324	16.0	4.7×10^{-2}
	HCA(II), $m = 2$	359	22.0	3.6×10^{-3}
	HCA(II), $m = 3$	411	30.4	8.3×10^{-4}
	HCA(II), $m = 4$	486	38.0	1.0×10^{-4}
Cube	ACA, $\varepsilon = 10^{-2}$	791	14.8	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-3}$	894	18.2	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-4}$	1034	22.6	1.8×10^{-2}
	ACA, $\varepsilon = 10^{-5}$	1202	27.4	1.8×10^{-2}
Cube	HCA(II), $m = 1$	364	12.9	1.8×10^{-1}
	HCA(II), $m = 2$	401	18.9	2.8×10^{-2}
	HCA(II), $m = 3$	471	27.0	3.9×10^{-3}
	HCA(II), $m = 4$	576	34.6	8.3×10^{-4}

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

M. Bebendorf, L. Grasedyck:
$\mathcal{H}$-Matrix Preconditioning for BEM,
in preparation.

S. Börm, L. Grasedyck:
Hybrid Cross Approximation,
$\Rightarrow$ Preprint soon at `http://www.mis.mpg.de`.

Next Winterschool on $\mathcal{H}$-matrices:
February 7th 2005 — February 12th 2005
MPI Leipzig, see `www.hmatrix.org`