Hierarchical Matrices

Wolfgang Hackbusch

Max-Planck-Institut für *Mathematik in den Naturwissenschaften*

Inselstr. 22-26, D-04103 Leipzig, Germany

wh@mis.mpg.de

http://www.mis.mpg.de/index.html

KAUST, May 9, 2016

Overview

- Introduction
- Model Example for Demonstration
- General Construction of Hierarchical Matrices
- Application to Boundary Element Methods (BEM)
- Application to Finite Element Methods (FEM)
- Matrix Equations
- Matrix-Valued Functions
- Application to Eigenvalue Problems
- $\mathcal{H}^2$-Matrices

1 Introduction

1.1 General Goal

Computations involving $n \times n$ matrices require up to $\mathcal{O}(n^3)$ operations.

Fully populated matrices need a storage of $O(n^2)$.

Goal of the hierarchical matrix technique:
storage and all matrix operations, in particular for full matrices, should be of almost linear cost (more precisely $O(n \log^* n)$).

This is not possible in general, but it holds for matrices of elliptic origin.
The results are only approximate (only $A * x$ is exact).

Already existing discretisation error $\varepsilon = O(n^{-\alpha}) \Rightarrow \log(1/\varepsilon) = O(\log n)$.

Typical fields of application:

- Boundary Element Method (BEM):

Formulation of homogeneous elliptic boundary value problems by integral equation formulations
$\Rightarrow$ System matrices are fully populated

- Finite Element Method (FEM):

Elliptic boundary value problems lead to sparse matrices A, but for instance A^{-1} is full. LU factors are partially filled.

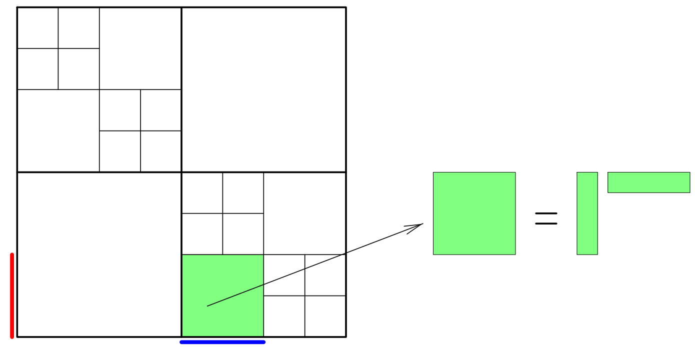
- Further Applications

Preview: How do $\mathcal{H}$-matrices look like?

- Decompose the matrix into suitable subblocks.
- Approximate the matrix in each subblock by a rank-k matrix

$$block = \sum_{i=1}^{k} a_i b_i^\top = AB^\top$$

(for suitably small local rank k). Illustration:



Two key questions:

- How large is the representation error?
More precisely: How does the local rank k correspond to the error of the matrix representation?
- How can the (approximate) matrix operations be performed such that

$$cost = O(n * \log^* n) \text{ ?}$$

1.2 Remarks concerning $\mathcal{R}_k$-Matrices

Let $M \in \mathbb{R}^{n\times m}$. We write $M \in \mathcal{R}_k$ if matrices

$$A \in \mathbb{R}^{n\times k} \quad \text{and} \quad B \in \mathbb{R}^{m\times k}$$

are given such that

$$M = AB^\top = \sum_{i=1}^{k} a_i b_i^\top .$$

Advantage: low storage, easy to multiply.

Disadvantage: $M', M'' \in \mathcal{R}_k \quad \Rightarrow \quad M' + M'' \in \mathcal{R}_{2k}.$

Remedy: For $M' \in \mathcal{R}_{k'}$ with $k' > k$, determine SVD $M' = \sum_{i=1}^{k'} \sigma_i u_i v_i^\top$. Replace all singular values σ_ℓ $(k < \ell \leq k')$ by zero:

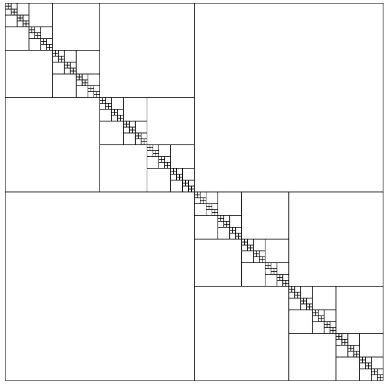
$$M = T_k M' = \sum_{i=1}^{k} \sigma_i u_i v_i^\top \in \mathcal{R}_k \quad \text{by } M := U\mathbf{\Sigma}V^\top$$

(T_k: truncation to rank k, best approximation).

Addition: $M' \oplus_{\mathcal{R}_k} M'' := T_k\left(M' + M''\right).$

2 Example for Demonstration

Let $n = 2^p$, $p = 0, 1, \ldots$

The construction of the $\mathcal{H}$-matrix format  is recursive:

For $n = 1$, A is a rank-1 matrix.
Otherwise, A is an $n \times n$ matrix of level p ($n = 2^p$):

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array} \right]$$

with

- A_{ij} are blocks of the size $\frac{n}{2} \times \frac{n}{2}$,

- A_{ii} ($i = 1, 2$) are $\mathcal{H}$-matrices (of level $p - 1$),

- A_{12}, A_{21} are rank-1 matrices (abbreviation: $R1$).

2.1 Complexity of the $\mathcal{H}$-Matrix Arithmetic

Dimension: $n = 2^p$, p : hierarchy level, $k := 1$.

. . . .

Number of blocks: $N_{\text{block}} = 3n - 2$

Storage: $S = n + 2n \log_2 n$

Matrix $\times$ vector: $N_{\text{MV}} = 4n \log_2 n - n + 2$

Matrix addition ($\oplus_{\mathcal{R}_1}$):

$$N_{H+H} = N_{H+R} = 17n \log_2 n + 39n - 38$$

Matrix-matrix multiplication ($\odot_{\mathcal{R}_1}$):

$$N_{R*R} = 3n - 1, \; N_{H*R} = 4n \log_2 n - n + 2,$$
$$N_{H*H} = \frac{25}{2} n \log_2^2 n + \frac{89}{2} n \log_2 n - 31n + 32$$

Matrix inversion:

$$N_{\text{inv}} = \frac{25}{2} n \log_2^2 n + \frac{55}{2} n \log_2 n - 69n + 70$$

LU decomposition

$$N_{LU} = \frac{21}{4} n \log_2^2 n + \frac{61}{4} n \log_2 n - 37\,(n - 1)\,.$$

2.1.1 Matrix-Matrix Multiplication

Three types of products are to be distinguished:

1. $R * R$ $\quad$ (R1-matrix $\times$ R1-matrix)

2. $R * H$ $\quad$ oder $H * R$

3. $H * H$

Type 1: $\left(ab^\top\right)\left(cd^\top\right) = (\alpha * a)\, d^\top$ mit $\alpha = b^\top c$ $\quad \Rightarrow N_{R1*R1}(p) = 3n - 1$ operations

---

Type 2: $H * \left(ab^\top\right) = (H * a)\, b^\top$ requires only one matrix-vector multiplication.

$\Rightarrow N_{H*R1}(p) = 4n \log_2 n - n + 2$ operations. Similar for $R * H$.

Type 3: $H * H$ is determined recursively:

$$\begin{aligned} H * H &= \begin{bmatrix} H & R \\ R & H \end{bmatrix} * \begin{bmatrix} H & R \\ R & H \end{bmatrix} \\ &= \begin{bmatrix} \underline{H * H} + R * R & H * R + R * H \\ R * H + H * R & \underline{\underline{H * H}} + R * R \end{bmatrix}. \end{aligned}$$

This yields the recursion

$$N_{H*H}(p) = 2N_{H*H}(p-1) + 25n \log_2 n + 32n - 32$$

with starting value $N_{H*H}(0) = 1$.

LEMMA: Multiplication of two $\mathcal{H}$-matrices requires

$$\frac{25}{2} n \log_2^2 n + \frac{89}{2} n \log_2 n - 31n + 32 \text{ operations.}$$

This operation is inexact because of truncation of sums.

2.1.2 Matrix Inversion

$$
\begin{aligned}
M^{-1} &= \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}^{-1} \\
&= \begin{bmatrix} M_{11}^{-1} + M_{11}^{-1} M_{12} S^{-1} M_{21} M_{11}^{-1} & -M_{11}^{-1} M_{12} S^{-1} \\ -S^{-1} M_{21} M_{11}^{-1} & S^{-1} \end{bmatrix},
\end{aligned}
$$

where

$$S = M_{22} - M_{21} M_{11}^{-1} M_{12}.$$

LU Decomposition

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix} \Leftrightarrow$$

(i) Compute L_{11} and U_{11} as factors of the LU decomposition of A_{11}.
(ii) Compute U_{12} from $L_{11}U_{12} = A_{12}$.
(iii) Compute L_{21} from $L_{21}U_{11} = A_{21}$.
(iv) Compute L_{22} and U_{22} as factors of the LU decomposition of $L_{22}U_{22} = A_{22} - L_{21}U_{12}$.

Solve steps (i) and (iv) recursively.

2.2 Concluding Remarks to the Introductory Case

At least, the rank 1 is to be replaced by a larger rank k. Moreover, in general, the simple format is to be replaced by a more refined format:

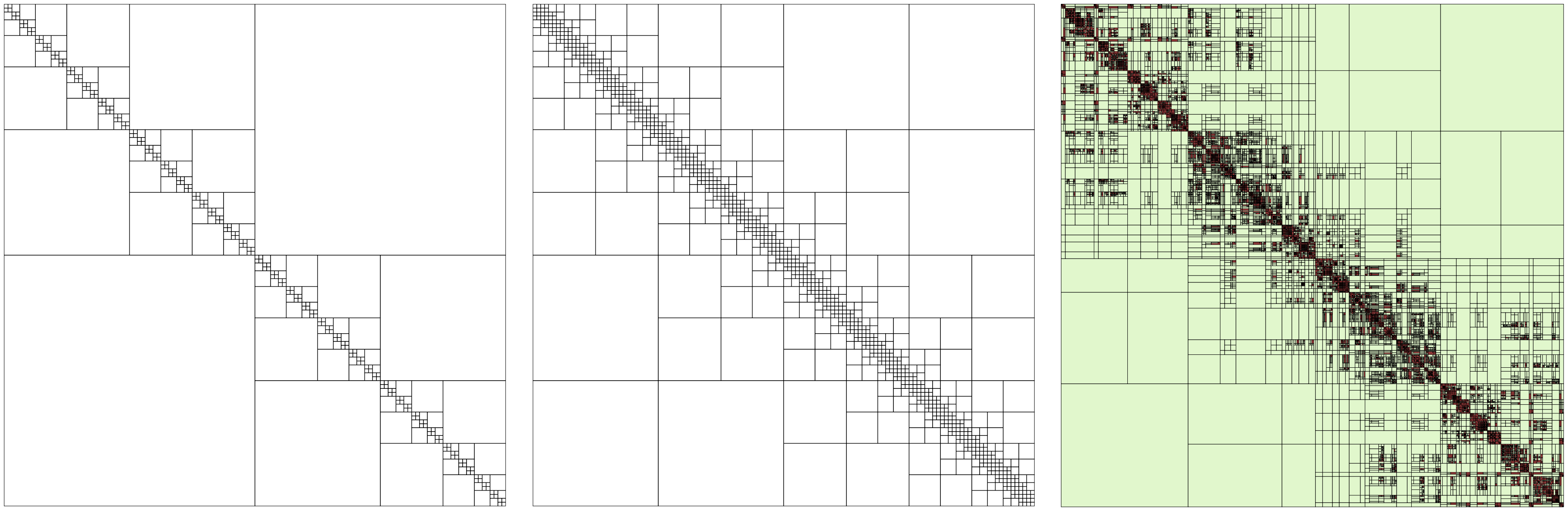
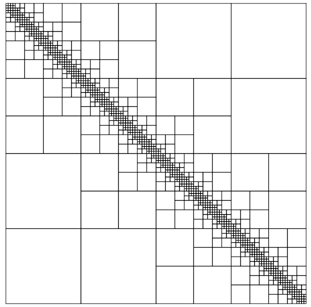


Figure 1: Left: simple block partition $\mathcal{H}_7$. Middle: admissible block partition. Right: partition for a real-life application of size 447488.

3 General Construction of Hierarchical Matrices

3.1 Partition of the Matrix



How to partition the matrix into subblocks?

I: index set of matrix rows,
J: index set of matrix columns.
Block: $b = \tau \times \sigma$ with $\tau \subset I$, $\sigma \subset J$.

Cluster Tree:
The cluster tree $T(I)$ contains a collection of subsets $\tau \subset I$ (similarly: $T(J)$).

Notation: $S(\tau)$ is the set of sons of a vertex τ.

Definition 3.1 *The Cluster tree $T = T(I)$ satisfies:*
(i) $I \in T$ (I is root of T),
(ii) If $\tau \in T$ is no leaf, $S(\tau)$ contains disjoint subsets of I such that

$$\tau = \bigcup_{\sigma \in S(\tau)} \sigma. \tag{1}$$

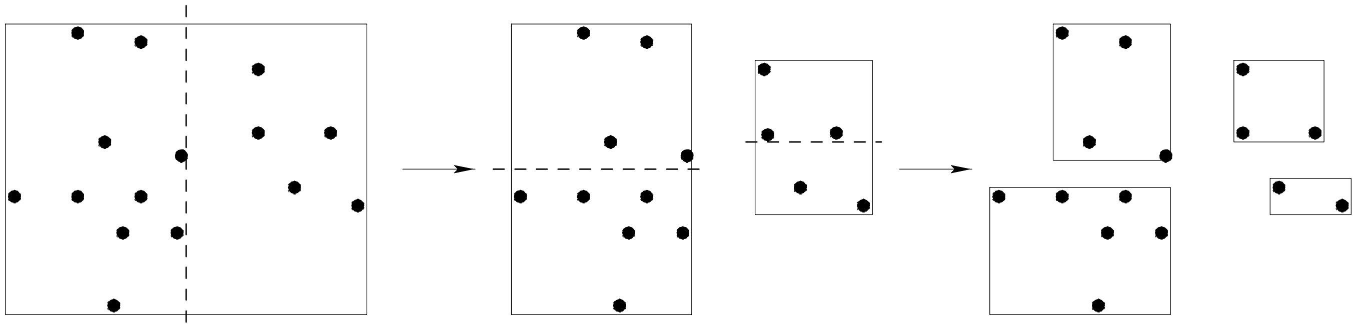
(iii) $\tau \in T$ is a leaf, if $1 \leq \#\tau \leq n_{\text{min}}$.

In the following $n_{\text{min}} = 1$ (but $n_{\text{min}} \approx 20$ is more practical).

3.2 General Construction of $T(I)$

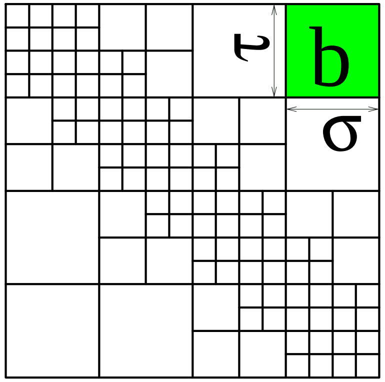
Let $\Omega \subset \mathbb{R}^d$ be a domain or a surface and $\mathbf{x}_i$ $(i \in I)$ the nodal points (e.g., corresponding to a basis function of a FEM discretisation).

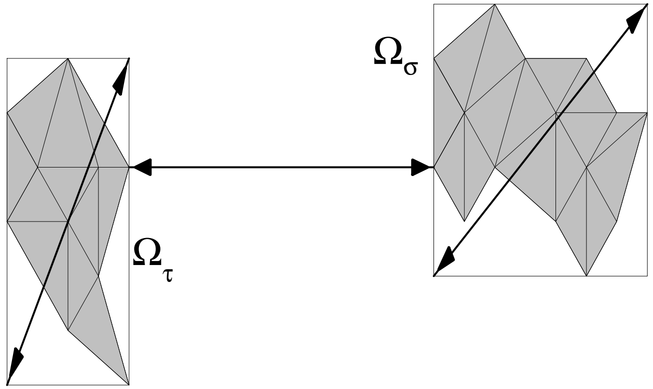
The practical performance uses bounding boxes and a recursive splitting of the longest side:



Block Cluster Tree $T(I \times J)$:

Collection of (small and large) blocks $b = \tau \times \sigma$ with $\tau \in T(I)$, $\sigma \in T(J)$.

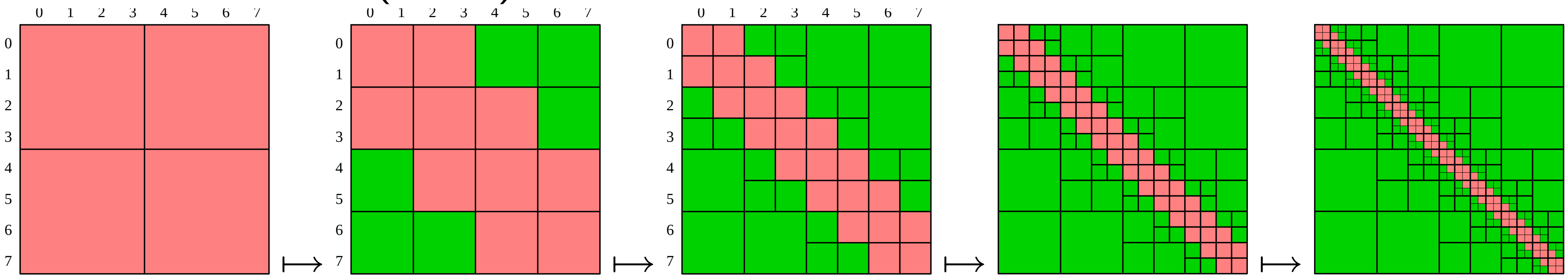




Criterion for selection: b as large as possible and admissible, i.e.,

$$\min\{\mathrm{diam}(\tau), \mathrm{diam}(\sigma)\} \leq \eta\, \mathrm{dist}(\tau, \sigma).$$

Levels of the tree $T(I \times J)$:



green blocks: admissible. Approximate by $\mathcal{R}_k$ matrices

red: non-admissible, but small enough. Use full matrix blocks.

About the admissibility condition

What is the optimal choice of block sizes w.r.t. storage cost?

Elliptic background:
behaviour of singularity functions, e.g., $\varkappa(x, y) = 1/\|x - y\|$

Such functions are **asymptotically smooth**, i.e.,

$$|\partial_x^\alpha \partial_y^\beta \varkappa(x, y)| \leq c_{\mathsf{as}}(\alpha + \beta)\,|x - y|^{-|\alpha|-|\beta|-s} \qquad \text{for } \alpha, \beta \in \mathbb{N}^d$$

with some $s \in \mathbb{R}$ and $c_{\mathsf{as}}(\nu) = C\,\nu!\ |\nu|^r\ \gamma^{|\nu|}$.

Then Taylor expansion shows:
a fixed accuracy can be obtained for all blocks satisfying

$$\min\{\mathrm{diam}(\tau), \mathrm{diam}(\sigma)\} \leq \eta\,\mathrm{dist}(\tau, \sigma)$$

with the *same* polynomial degree ($\rightarrow$ same rank).

3.3 Matrix Operations

Matrix-vector multiplication: $M \in \mathbb{R}^{I\times J}$, $x \in \mathbb{R}^{J}$, $y \in \mathbb{R}^{I}$.

procedure *MVM*(y, M, x, b);
if $b = \tau \times \sigma \in P$ **then** $y|_\tau := y|_\tau + M|_b \cdot x|_\sigma$
else for all $b' \in S(b)$ **do** *MVM*(y, M, x, b');

Here, P denotes the partition.

The call

$$\mathit{MVM}\,(y, M, x, I \times J)$$

produces $y := y + Mx$.

4 Application to BEM

Example: $$(\mathcal{A}u)(x) := \int_0^1 \log|x-y|\, u(y)dy \qquad \text{for } x \in [0,1].$$

Discretisation: collocation with piecewise constant elements in

$$[x_{i-1}, x_i],\ x_i = ih, \qquad i = 1, \ldots, n,\ h = 1/n,$$

Midpoints $x_{i-1/2} = (i-1/2)h$ are the collocation points:

$$A = (a_{ij})_{i,j=1,\ldots,n} \qquad \text{with } a_{ij} = \int_{x_{j-1}}^{x_j} \log\left|x_{i-1/2} - y\right| dy.$$

Replace the kernel function $\kappa(x,y) = \log|x-y|$ in a certain range of x, y by an approximation $\tilde{\kappa}(x,y)$ of separable form

$$\tilde{\kappa}(x,y) = \sum_{\iota \in J} X_\iota(x) Y_\iota(y).$$

$$\tilde{\kappa}(x,y) = \sum_{\iota\in J} X_\iota(x) Y_\iota(y).$$

Simple choice: Taylor's formula applied with respect to y:

$$\begin{aligned} J &= \{0,1,\ldots,k-1\}, \\ X_\iota(x) &= \text{derivatives of } \kappa(x,\cdot) \text{ evaluated at } y = y^*, \\ Y_\iota(y) &= (y-y^*)^\iota. \end{aligned}$$

The kernel $\kappa(x,y) = \log|x-y|$ leads to the error estimate

$$|\kappa(x,y) - \tilde{\kappa}(x,y)| \le \frac{|y-y^*|^k/k}{(|x-y^*| - |y-y^*|)^k} \quad \text{for} \quad |x-y^*| \ge |y-y^*|.$$

If κ is replaced by $\tilde{\kappa}$, the integral $a_{ij} = \int_{x_{j-1}}^{x_j} \kappa(x_{i-1/2}, y)dy$ becomes

$$\tilde{a}_{ij} = \sum_{\iota\in J} X_\iota(x_{i-1/2}) \int_{x_{j-1}}^{x_j} Y_\iota(y)dy \qquad ((i,j)\in b). \qquad (*)$$

$(*)$ describes a matrix block $\tilde{A}|_b$.
Each term of the sum in $(*)$ is an $\mathcal{R}_1$ matrix $ab^\top$ with

$$a_i = X_\iota(x_{i-1/2}), \quad b_j = \int_{x_{j-1}}^{x_j} Y_\iota(y)dy.$$

Since $\#J = k$, the block $\tilde{A}|_b$ belongs to $\mathcal{R}_k$.

Furthermore, one can check that

$$|\kappa(x,y) - \tilde{\kappa}(x,y)| \leq \frac{1}{k}\left(\frac{1}{2}\right)^k, \qquad \|A - \tilde{A}\|_\infty \leq 2^{-k}/k.$$

Discretisation error $h^{\varkappa}$, where the step size h is related to $n = \#I$ by $h \sim \frac{1}{n}$. k should be chosen such that

$$2^{-k} \sim \left(\frac{1}{n}\right)^{\varkappa}.$$

Hence,

$$k = O(\log n)$$

is the required rank.

NOTE: a) The construction of the cluster and block-cluster tree is automatic (black box) and fast. Even refinements with form-regular elements are allowed. b) Similarly, the construction of the approximation $\tilde{A}$ is black-box like (usually by interpolation instead of Taylor expansion).

Alternative Construction of the BEM Matrix: Cross Approximation

For all admissible blocks evaluate k rows and k columns (k *crosses*).

Determine the unique rank-k matrix interpolating the cross data.

Cost for an $n \times m$ block: $\leq k(n + m)$ evaluations.

Adaptive choice of crosses $\rightarrow$ ACA (adaptive cross approximation)

Advantage: Method uses only matrix data, no information about the kernel function etc. necessary.

5 Application to FEM

REMARK A FEM system matrix is an $\mathcal{H}$-matrix (without any approximation error).
Proof: Non-trivial blocks $= 0$.

REMARK For a uniformly elliptic differential operator with L^∞-coefficients in a Lipschitz domain, the inverse of the FEM-matrix can be exponentially well approximated by an hierarchical matrix.

Literature:
Bebendorf–Hackbusch, Numer. Math. 95 (2003) 1-28
Faustmann–Melenk–Praetorius, Numer. Math. 131 (2015) 615-642.

Analytical Background

Boundary value problem:

$$\mathrm{div}(A(x)\,\mathrm{grad}\,u) = f(x), \quad x \in \Omega \subset \mathbb{R}^d, \text{ with}$$
$$A \in L^\infty(\Omega), \text{ eigenvalues} \in [c', c''], \; c' > 0.$$

$X, Y \subset \Omega$ admissible subsets, i.e., $\min\{\mathrm{diam}(X), \mathrm{diam}(Y)\} \leq \eta \; \mathrm{dist}(X, Y)$.

Then the Green function $G(x, y)$ admits an expansion

$$G(x, y) = \sum_{\nu=1}^{\infty} g'_\nu(x) g''_\nu(y) \qquad \text{for } x \in X, y \in Y,$$

which is **exponentially** convergent.

6 $\mathcal{H}$-LU iteration

Linear system of equations:

$$Ax = b.$$

Determine the LU decomposition of A by using hierarchical factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$.

Since $L_{\mathcal{H}}U_{\mathcal{H}}$ is very close to A, it is a very good 'preconditioner'; i.e., the iteration

$$x^{m+1} = x^m - (L_{\mathcal{H}}U_{\mathcal{H}})^{-1}(Ax^m - b)$$

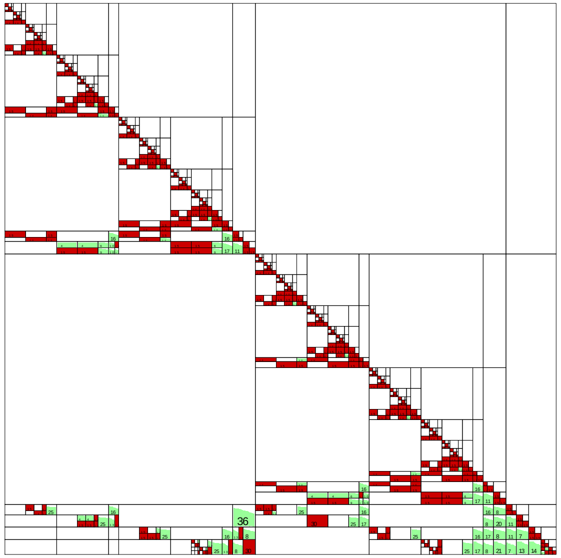
is a very fast iteration.

One can prove:
If the inverse matrix can be approximated by hierarchical matrices, then also the LU factors do so.

$\mathcal{H}$-LU iteration for sparse matrices

The partition of the matrix can be modified so that it corresponds to the nested dissection technique of A. George (1973).
Then sparsity of A is partially inherited by $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$!

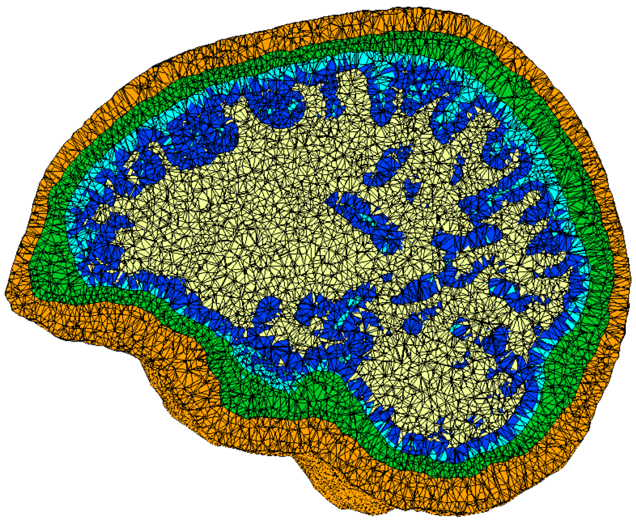
Example of a factor L:



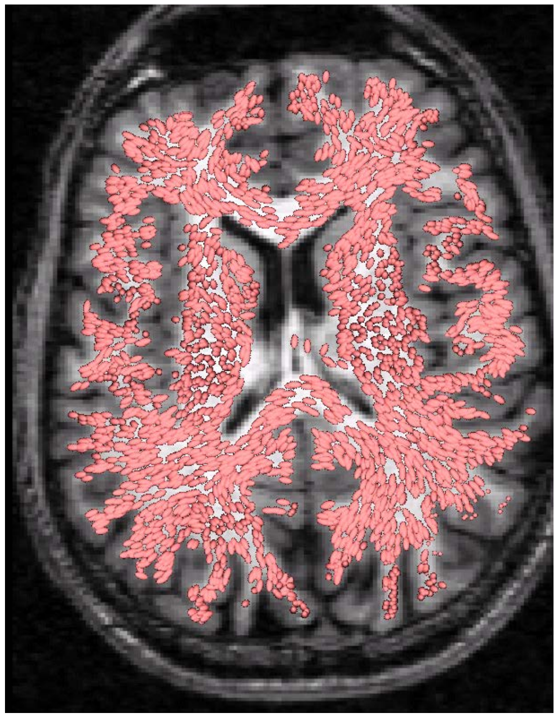
Using the graph distance of $G(A)$ instead of the Euclidean distances, one obtains an **algebraic iteration** $\rightarrow$ perfect for blackbox applications.

EXAMPLE (inverse problem, Wolters-Grasedyck-Hackbusch, 2004):
Given: electric/magnetic field (EEG,MEG) at $\approx$ 400 sensor positions on the head surface.
What is the current distribution in the brain ? Where are the sources ?

PDE: $-\text{div}\,\sigma(x)\nabla u(x) = f(x), \quad x \in \Omega \subset \mathbb{R}^3$, $\partial_n u = 0$ on $\partial\Omega$.
The boundary value has to be solved for $\approx$ 400 right-hand sides



Triangulation with
$N = 147287$ tetraeder



conductivity σ

- Galerkin discretisation $\rightsquigarrow Ax = b$

- The system has to be solved for $\approx$ 400 right-hand sides b

- Stopping criterion: $\|Ax - b\| / \|b\| \leq 10^{-8}$

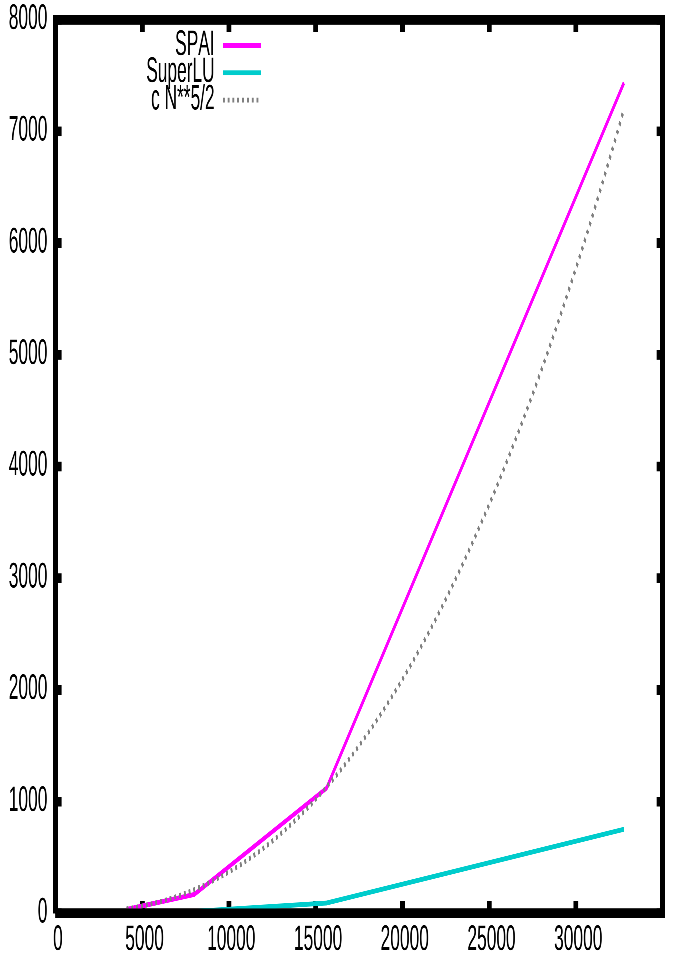
- Machine: SUNFire, 900 MHz, single processor

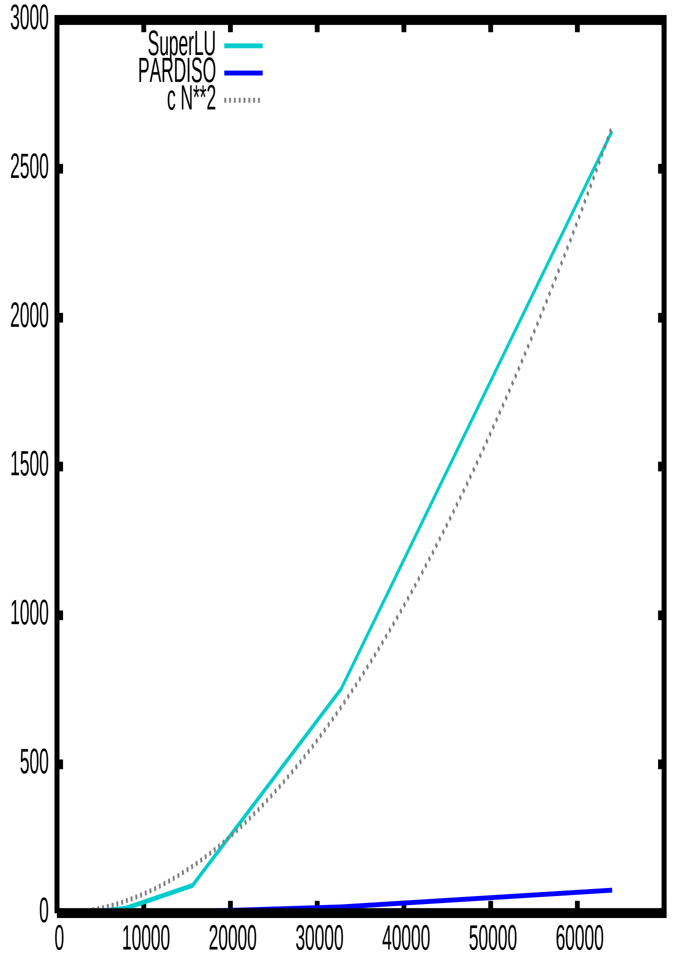
	Pardiso*	$LU_{\mathcal{H}},\ \varepsilon = 10^{-6}$	PEBBLES†
Setup	237	468	13
Solve	2.4	1.0	10
Total	1197	868	4013

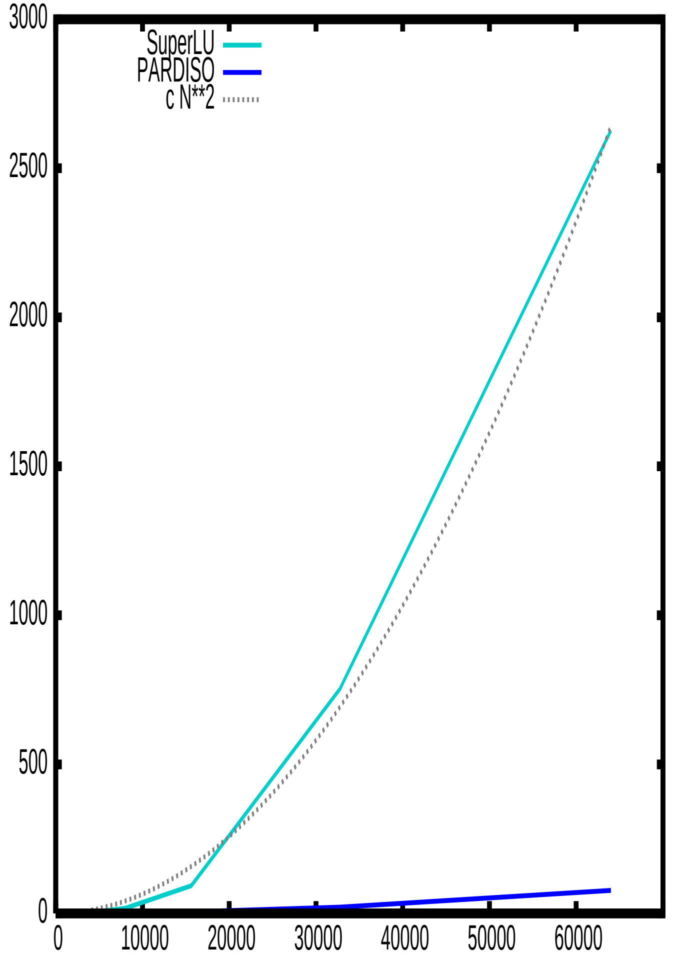
*Pardiso (Schenk & Co)

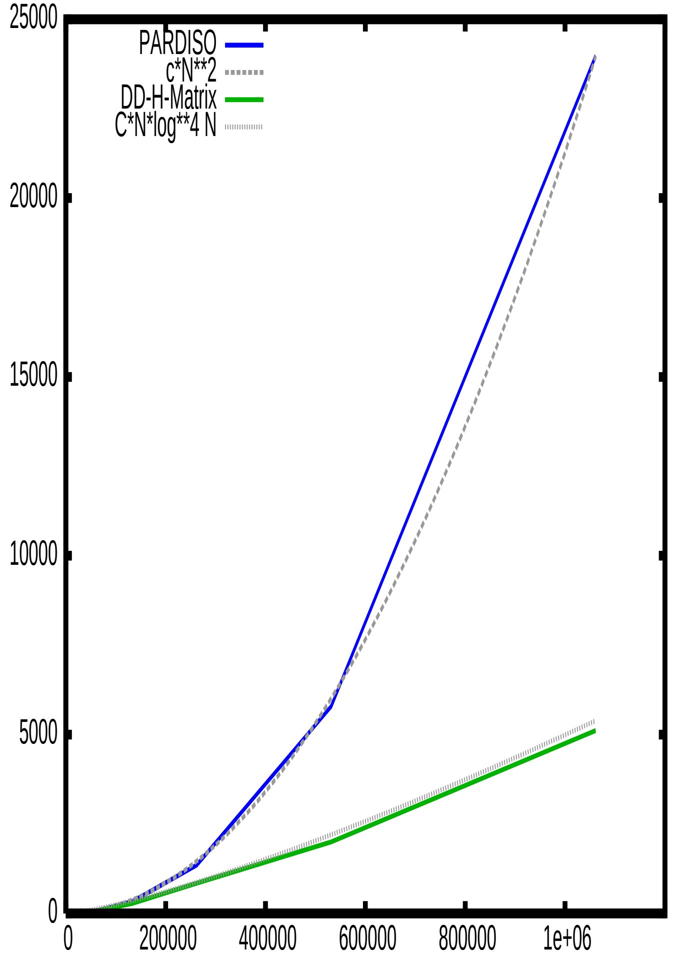
†PEBBLES (Langer/Haase)

Comparisons by a simple test example






SuperLU
PARDISO
c N**2
0
500
1000
1500
2000
2500
3000
0
10000
20000
30000
40000
50000
60000


PARDISO
c*N**2
DD-H-Matrix
C*N*log**4 N
0
5000
10000
15000
20000
25000
0
200000
400000
600000
800000
1e+06

Numerical examples $\longrightarrow$ Lecture of R. Kriemann

Parallelisation $\longrightarrow$ Lecture of R. Kriemann

7 Matrix Equations

$$\begin{aligned}
&\text{Lyapunov:} & AX + XA^\top &= C \\
&\text{Sylvester} & AX - XB &= C \\
&\text{Riccati:} & AX + XA^\top - XFX &= C
\end{aligned}$$

Given: A, B, C, F; desired matrix-valued solution: X.

Applications: optimal control problems for elliptic / parabolic pdes.

- Low rank $C, F \Rightarrow$ low rank X

- $\mathcal{H}$-matrix C, low rank $F \Rightarrow \mathcal{H}$-matrix X

Computation via $\mathcal{H}$-arithmetic, possibly combined with multigrid methods.

Matrix-Riccati Equation

$$A^\top X + XA - XFX + G = O \qquad (A < O).$$

Lemma 7.1 *The solution* X *satisfies*

$$X = -(M^\top M)^{-1} M^\top N,$$

where

$$\begin{bmatrix} M & N \end{bmatrix} := \mathrm{sign}\left(\begin{bmatrix} A^\top & G \\ F & -A \end{bmatrix} \right) - \begin{bmatrix} I & O \\ O & I \end{bmatrix}.$$

Lemma 7.2 *Assume that* $\Re e \lambda \neq 0$ *for all eigenvalues* $\lambda \in \sigma(S)$.

Start: $S^{(0)} := S$. *Then the iteration*

$$S^{(i+1)} := \frac{1}{2}\left(S^{(i)} + \left(S^{(i)}\right)^{-1} \right)$$

converges quadratically to $\mathrm{sign}(S)$.

Example of a matrix-Riccati equation

by L. Grasedyck: Computing **70** (2003) 121-165

Choice of A by $A = \Delta_h$ (1D-Laplacian).

The following table shows the relative error $\|\tilde{X} - X\|_2 / \|X\|_2$.

	$n = 101$	256	1024	65 536
$k = 1$	8.8_{10}-3	1.5_{10}-1	1.3_{10}-1	-
$k = 2$	2.4_{10}-4	2.6_{10}-4	4.2_{10}-4	6.7_{10}-4
$k = 4$	7.7_{10}-8	9.1_{10}-8	1.1_{10}-7	6.2_{10}-7
$k = 6$	1.9_{10}-10	3.7_{10}-10	2.4_{10}-10	1.7_{10}-9
Number of iterations	12	14	17	26
time* [sec]	2.2	8.5	67	18263

*) k=2, Sun Quasar 450 MHz

In the last case, the (full) matrix X has $4,294,967,296$ entries.

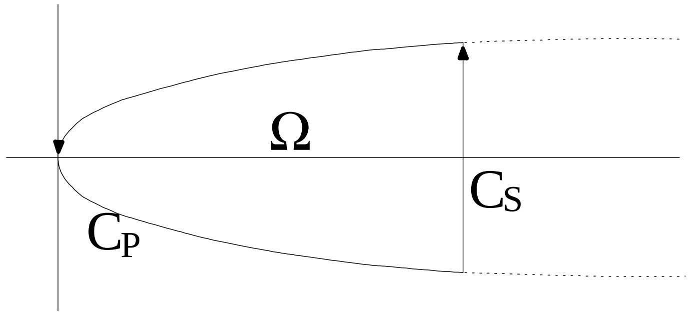
8 Matrix-Valued Functions $f(A)$

EXAMPLE: Matrix-exponential function e^{-tA}.

Cauchy-Dunford representation

$$e^{-tA} = \frac{1}{2\pi i} \int_\Gamma e^{-zt} (zI - A)^{-1} \, dz$$

using a parabola Γ :



After parametrisation and quadrature:

$$T_N(t) := \sum_{\ell=-N}^{N} \gamma_\ell e^{-\alpha_\ell t} (z_\ell I - A)^{-1}, \qquad z_\ell \in \Gamma.$$

Error estimate for $t \geq t_0 > 0$:

$$\left\| T_N(t) - e^{-tA} \right\| \lesssim e^{-cN^{2/3}}.$$

$\Rightarrow N \sim \log n \;\Rightarrow$ Total cost: $O(n \log^* n)$.

See: Gavrilyuk-Hackbusch-Khoromskij, Numer. Math. 92 (2002) 83-111

Second Approach to $\exp(-tA)$ **— 'Halving Method'**

If $\|tA\| \leq 1$, use Taylor:

$$\exp(-tA) \approx I - tA + \ldots + (-tA)^{n-1}/\,(n-1)!$$

Accuracy: $error \leq 1.72/n!$

Cost: $n-2$ matrix-matrix multiplications

If $\|tA\| > 1$:

$$\exp(-tA) = \exp(-\tfrac{t}{2}A) * \exp(-\tfrac{t}{2}A).$$

Cost:

1 matrix multiplication per recursion step

$O(\log_2(\|tA\|))$ recursion steps.

9 Applications to Eigenvalue Problems

– LR and QR methods using a special partition.

– Vector iteration and Krylov methods.

– Preconditioned inverse iteration.

– Bisection method (based on Sylvester's inertia law).

– Divide-and-conquer method.

– Eigenvalue distribution: LU decomposition of $A - (\lambda_0 + i\mu)\, I$ allows the computation of the determinant of the inverse. Derived from that one obtains, e.g., the spectral density

$$\sum_{\lambda \in \sigma(A)} \frac{\mu}{|\lambda - \lambda_0|^2 + \mu^2}.$$

– Spectral projection onto the spectral part in $int(\Gamma)$:

$$A_{\text{proj}} = \frac{1}{2\pi i} \int_\Gamma z\, (zI - A)^{-1}\, dz.$$

– Hierarchical AMLS (automated multi-level substructuring).

10 $\mathcal{H}^2$-Matrices

Two hierarchies are involved:

1. Hierarchy given by the cluster tree T.

2. The involved rank-k-matrices do not use arbitrary row and column vectors, but vectors from special subspaces V_τ $(\tau \in T)$, i.e., the matrix blocks $A|_{\tau \times \sigma}$ belong to tensor spaces $V_\tau \otimes V_\sigma$

3. The basis of V_τ is connected with the bases of $V_{\tau'}$ for $\tau' \in S(\tau)$. This leads to hierarchically defined bases: $V_\tau|_{\tau'} \subset V_{\tau'}$.

Since, in the end, the bases need not be stored directly, the log-factor disappears:

$$storage(A),\ cost(A * x),\ cost(A + B),\ cost(A * B) = O(n)$$

and smaller constants (see S. Börm 2010).

11 Literature

- W. Hackbusch: Hierarchical Matrices - Algorithms and Analysis. SSCM 49, Springer 2015.

- Börm, S.: Efficient Numerical Methods for Non-local Operators. EMS, Zürich 2010

- Bebendorf, M.: Hierarchical Matrices, Lect. Notes Comput. Sci. Eng. **63**. Springer, Berlin 2008

- W. Hackbusch: Iterative Solution of Large Sparse Systems of Equations. 2nd ed., Springer 2016 (Chapter 13 and Appendix D)

- *Low-rank approximations for tensor-structured data:* W.Hackbusch: Tensor Spaces and Numerical Tensor Calculus. Springer 2012

- Commercial software: , see `http://www.hlibpro.com/`