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/scicomp/hackbusch_e.html

Harrachov, August 20, 2007

1 Introductory Remarks

Aim

- Treatment of large-scale linear systems of equations is a common need in modern computations

Large-scale systems: size $n = 10^5, 10^6$ or larger, depending on the available storage size.

- The use of matrices and matrix operations lead in general to difficulties

Storage of $O(n^2)$ for fully populated matrices is usually not available.

Usual approach: Explicit use of matrices avoided, instead indirect matrix-vector multiplications (FFT, sparse matrices).

Here: Direct representation of matrices (dense matrices included).

Remark: Analysis vs. Linear Algebra

Traditionally, Analysis and Linear Algebra have different viewpoints concerning topology.

Example: In Analysis the set of functions is immediately restricted to certain subsets of different smoothness: L^2, C, C^k etc. A tool like a finite Taylor series can only be applied to the subset C^k.
In Linear Algebra, algorithms are usually required to work for *all* matrices (or symmetric or pos. def. matrices, etc.).

For large-scale problems, matrices are discretisations of operators. Hence, the topology of Functional Analysis is needed.

Consequence: Algorithms are considered to work for matrices with sufficient "smoothness".

Remark: Approximation

- Matrices arise after a discretisation process. Therefore, a further approximation error of similar (or smaller) size does not matter.

- Under certain "smoothness conditions" $n \times n$-matrices can be approximated by $O(n)$ or $O(n \log^* n)$ data ($\rightarrow$ N-term approximation with $N = O(n), O(n \log^* n)$).

TASK: One has to construct "data-sparse" representations of matrices involving only N data.

A typical size is

$$N = O(n \cdot \log n \cdot \log^d \tfrac{1}{\varepsilon}),$$

d: spatial dimension, ε: accuracy of the approximation.
If $\varepsilon \approx n^{-const}$, then $\log^d \frac{1}{\varepsilon} = O(\log^d n)$.

Remark: Matrix Operations

Low storage cost for matrices is only one aspect. The data-sparse representation must also support the relevant operations:

- matrix-vector multiplication
- transposition $A \to A^\top$
- matrix-matrix addition
- matrix-matrix multiplication
- matrix inversion
- LU decomposition

The results may be again approximations! Cost: $O(n \log^* n)$.

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 matrices

■ Finite Element Method (FEM):

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

Sometimes Schur complements

$$A_{11} - A_{12}\, A_{22}^{-1}\, A_{21}$$

are needed, which are also full.

■ Further Applications: matrix equations, matrix functions

2 Construction of Hierarchical Matrices

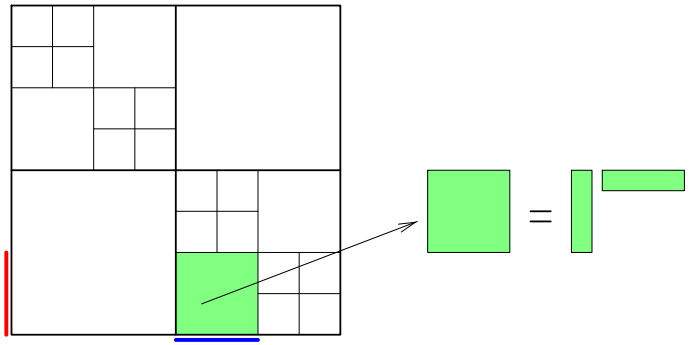
- Decompose the matrix into suitable subblocks.

- Approximate the matrix in each subblock by a rank-k-matrix*

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

(for suitably small local rank k).

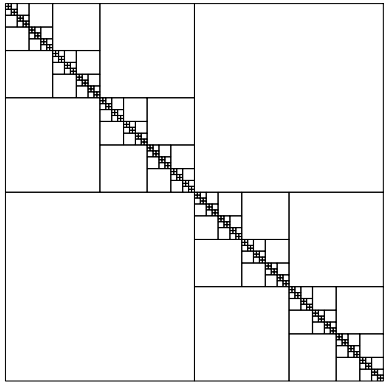
Illustration:



*k is upper bound. The true rank may be smaller.

Example for Demonstration

Let $n = 2^p$, $p = 0, 1, \ldots$ The $\mathcal{H}$-matrix format is chosen as follows:



All subblocks are filled by rank-k-matrices (here $k = 1$).

- number of blocks: $3n - 2$,
- storage cost: $n + 2n \log_2 n$,
- cost of matrix-vector multiplication: $4n \log_2 n - n + 2$.

Matrix Addition

Difficulty:
Addition of two rank-k submatrices yields rank $2k$.

Remedy:
Truncation to rank k (via SVD) yields a result in the same $\mathcal{H}$-matrix format.

Notation:
$A \oplus_k B$ is the true sum truncated to rank k.

- Cost for Rank-1-addition $\oplus_1$ is $18n \log_2 n + O(n)$.

Matrix-Matrix Multiplication

Recursion:

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

- The approximate multiplication of two $\mathcal{H}$-matrices requires

$$13n \log_2^2 n + 65n \log_2 n - 51n + 52$$

operations.

Matrix Inversion

The (exact) inverse of A is

$$\begin{bmatrix} A_{11}^{-1} + A_{11}^{-1} A_{12} S^{-1} A_{21} A_{11}^{-1} & -A_{11}^{-1} A_{12} S^{-1} \\ -S^{-1} A_{21} A_{11}^{-1} & S^{-1} \end{bmatrix}$$

with the Schur complement $S = A_{22} - A_{21} A_{11}^{-1} A_{12}$.

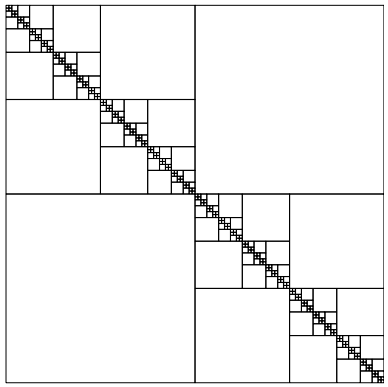
- The approximate inversion of an $\mathcal{H}$-matrix requires

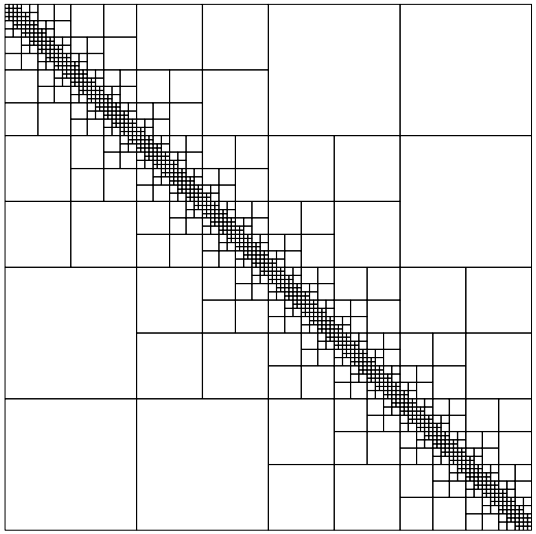
$$13n \log_2^2 n + 47n \log_2 n - 109n + 110 \text{ operations,}$$

- cost of approximate LU decomposition: $\frac{11}{2} n \log_2^2 n + 25n \log_2 n - 28n + 28$.

Remarks to the Introductory Example

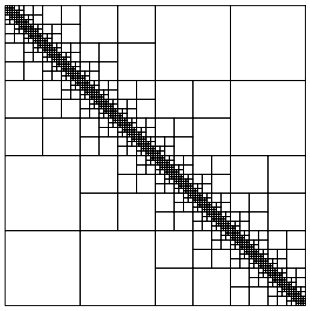
At least, the rank 1 is to be replaced by a larger rank k.

Moreover, in general, the simple format  is to be replaced by a more refined format like



General Construction of Hierarchical Matrices

Partition of the Matrix



How to partition the matrix in 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: $T(J)$).

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.,

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

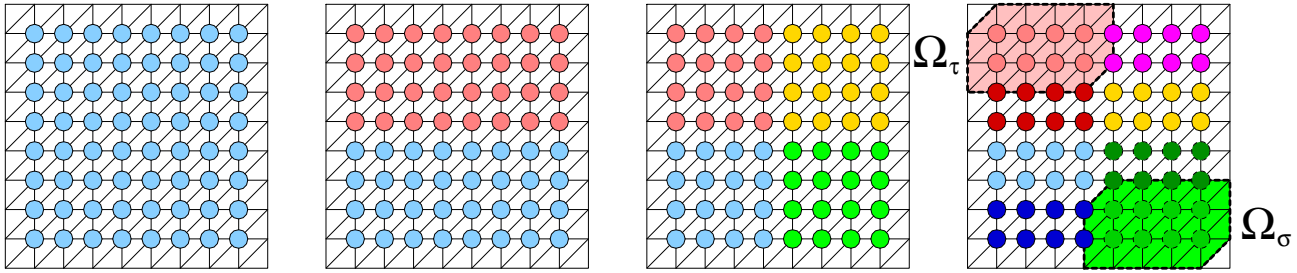
Cluster Tree

I: index set containing the row indices i of the matrix $A = \left(A_{ij}\right)$.

We partition I recursively into (e.g.) two subsets.

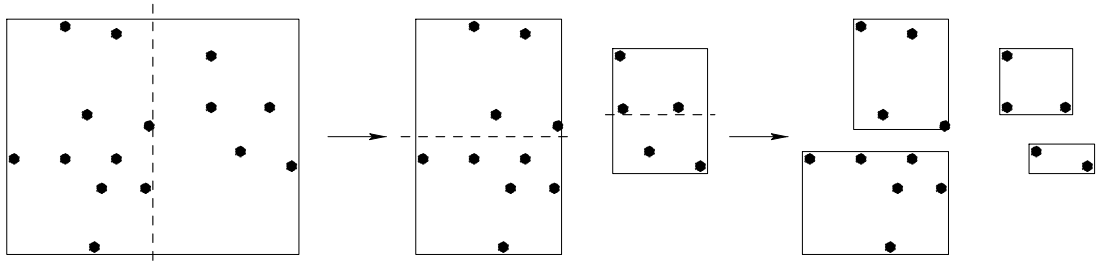
This process ends if the subsets of I have a sufficiently small cardinality.

The resulting tree $T(I)$ is called the cluster tree.



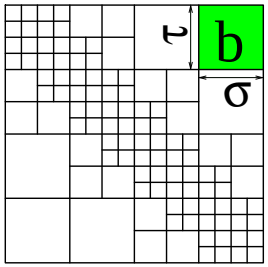
REMARK: For usual discretisations, an index $i \in I$ is associated to an nodal point $x_i \in \mathbb{R}^d$ or the support $\mathrm{supp}(\phi_i) \subset \mathbb{R}^d$ of a basis function ϕ_i.

The practical performance uses bounding boxes:

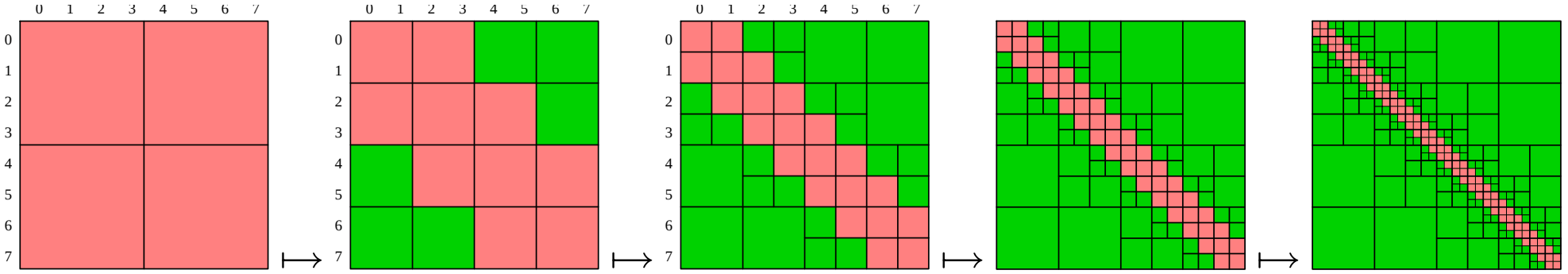


Block-Cluster Tree

NOTATION: $T(I \times J)$ is the block-cluster tree. Elements: blocks $b = \tau \times \sigma$.



Let $\tau \times \sigma \in T(I \times J)$ be a block ($\Longrightarrow \tau \in T(I)$, $\sigma \in T(J)$).
$\tau', \tau'' \in T(I)$ sons of τ, i.e., $\tau = \tau' \cup \tau''$. Similarly, $\sigma', \sigma'' \in T(J)$ sons of $\sigma \in T(J)$. Then the four sons of $\tau \times \sigma \in T(I \times J)$ are $\tau' \times \sigma'$, $\tau' \times \sigma''$, $\tau'' \times \sigma'$, $\tau'' \times \sigma'' \in T(I \times J)$. If either τ of σ is a leaf, $\tau \times \sigma$ is not further partitioned.

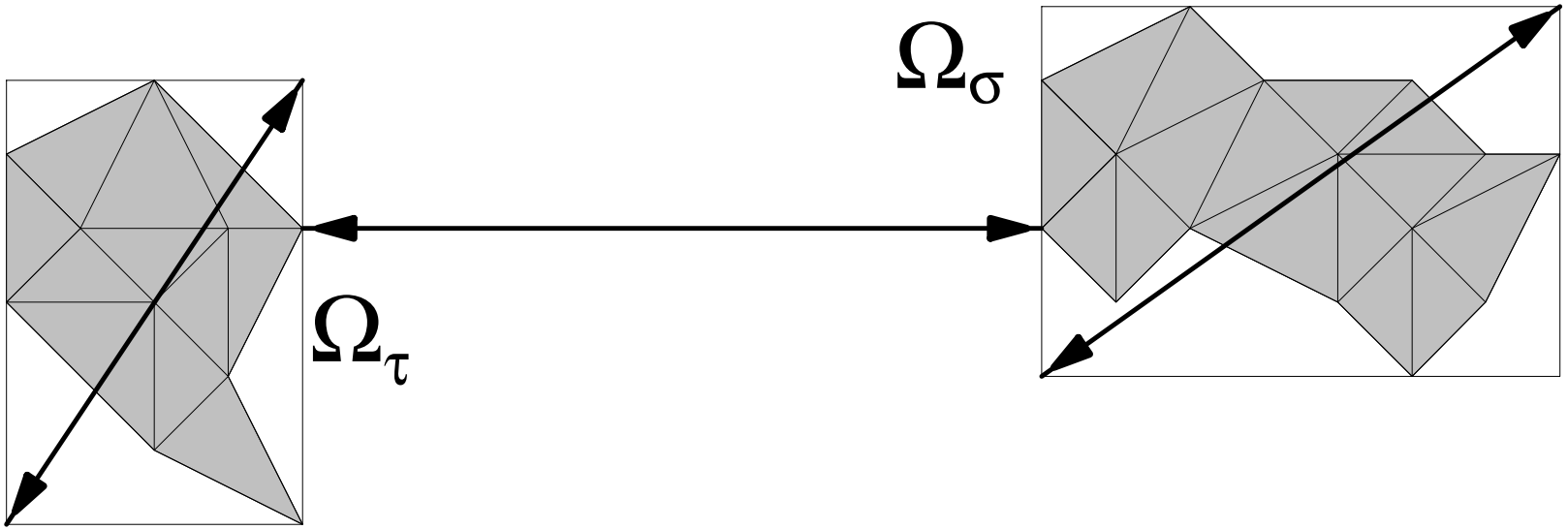


green blocks: admissible, red: non-admissible

DEFINITION (admissible block) Fix some $\eta > 0$. A block $\tau \times \sigma \in T(I \times J)$ is called admissible if

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

or $\tau \times \sigma$ is a leaf. $\tau \times \sigma \in T(I \times J)$ is called maximally admissible if the father of $\tau \times \sigma$ is non-admissible.



DEFINITION (Partition P): $P \subset T(I \times J)$ is defined by: 1) different $b \in P$ are disjoint, 2) their union $\bigcup_{b \in P} p = I \times J$ is complete, 3) they are maximally admissible.

3 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

$$\tilde{\kappa}(x,y) = \sum\nolimits_{\iota \in K} X_\iota(x) Y_\iota(y).$$

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

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

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

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

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

Let b be a block and restrict i, j in $(*)$ to b. Then $(*)$ describes a block matrix $\tilde{A}|_b$. Each term of the sum in $(*)$ is an rank-*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 $\#K = k$, the block $\tilde{A}|_b$ is of rank-k type.

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 $h \sim \frac{1}{n}$. Hence k should be chosen such that

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

Hence,

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

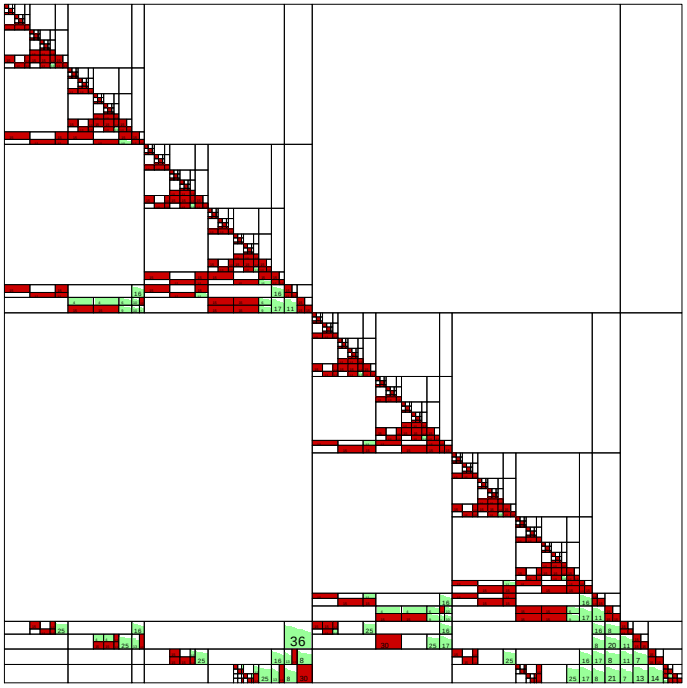
is the required rank.

NOTE: a) The construction of the cluster and block-cluster tree is automatic (black-box).
b) Similarly, the construction of the approximation $\tilde{A}$ is black-box (usually by interpolation instead of Taylor expansion).

4 Application to FEM

REMARK: a) A FEM system matrix is an $\mathcal{H}$-matrix. Non-trivial blocks $= 0$.
b) For a uniformly elliptic differential operator with L^∞-coefficients, the inverse of the FEM-matrix can be exponentially well approximated by an $\mathcal{H}$-matrix [Bebendorf - Hackbusch 2003].

When solving a linear system of equations $Ax = b$, one can make use of the LU decomposition. The particular advantage of the LU decomposition for sparse matrices A is that the factors L and U contain many zero block (fill-in is not complete!). Example of an factor L :

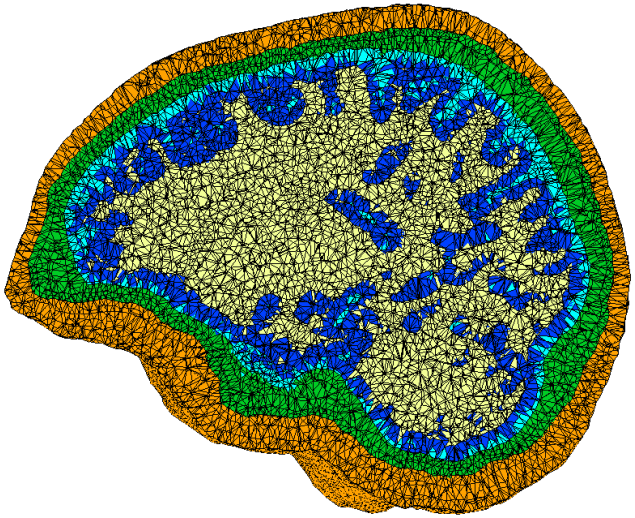


EXAMPLE (inverse Problem):

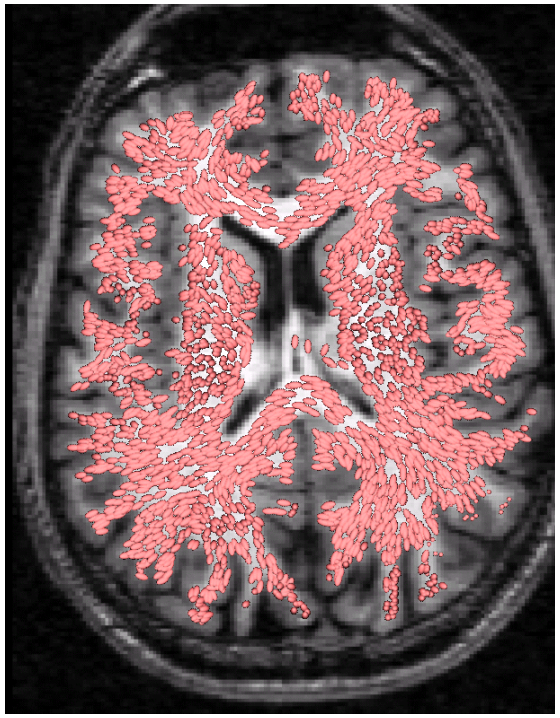
Given: electric/magnetic field at ≈ 400 sensor positions on the head surface. What is the current distribution in the brain ? Where are the sources (epileptic fit) ?

PDE: $-\mathrm{div}\,\sigma(x)\nabla u(x) = f(x), \quad x \in \Omega \subset \mathbb{R}^3$, $\partial_n u = 0$ on $\partial\Omega$.

Ω and $\sigma(x)$ determined from EEG,MEG. The boundary value problem has to be solved for ≈ 400 right-hand sides.



Triangulation with
$N = 147287$ tetraeder



conductivity σ

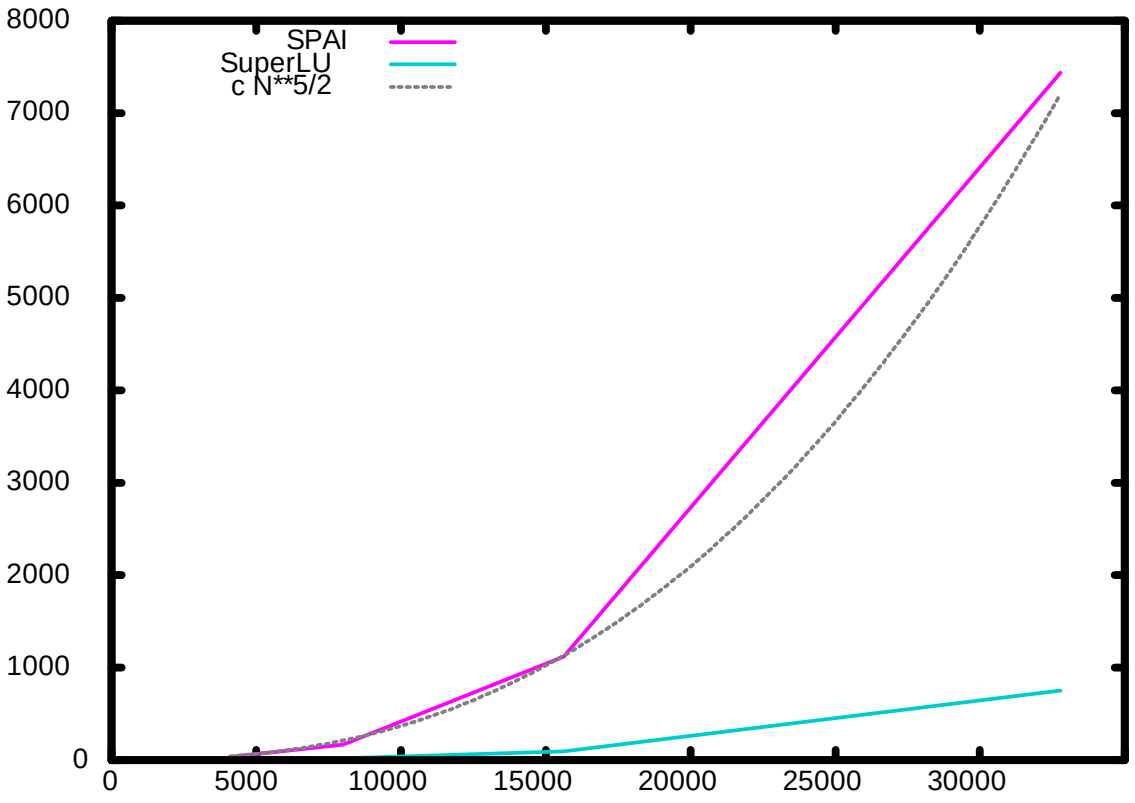
- Galerkin discretisation $\rightsquigarrow Ax = b$
- The system has to be solved for ≈ 400 right-hand sides b
- Stopping criterion: $\|Ax - b\| / \|b\| \leq 10^{-8}$
- Machine: SUNFire, 900 MHz, single processor

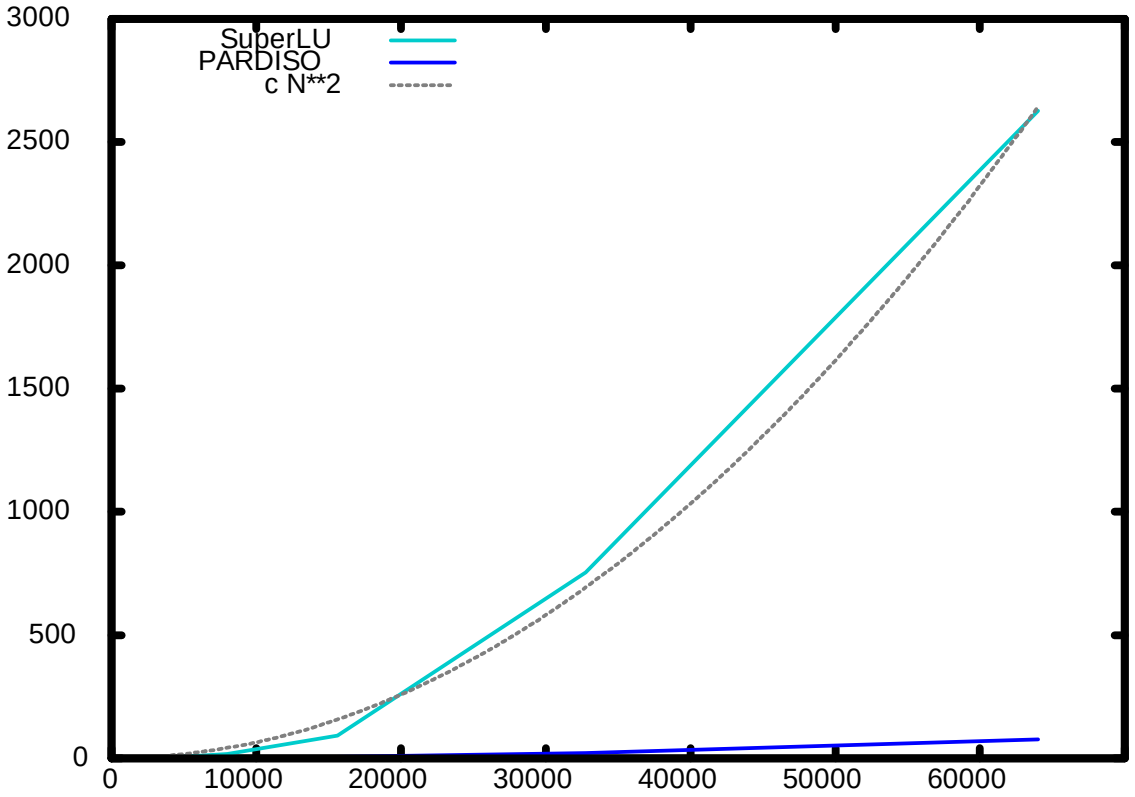
	Pardiso†	$\mathrm{LU}_{\mathcal{H}}$, $\varepsilon = 10^{-6}$	PEBBLES‡
Setup	237	468	13
Solve	2.4	1.0	10
Total	1197	868	4013

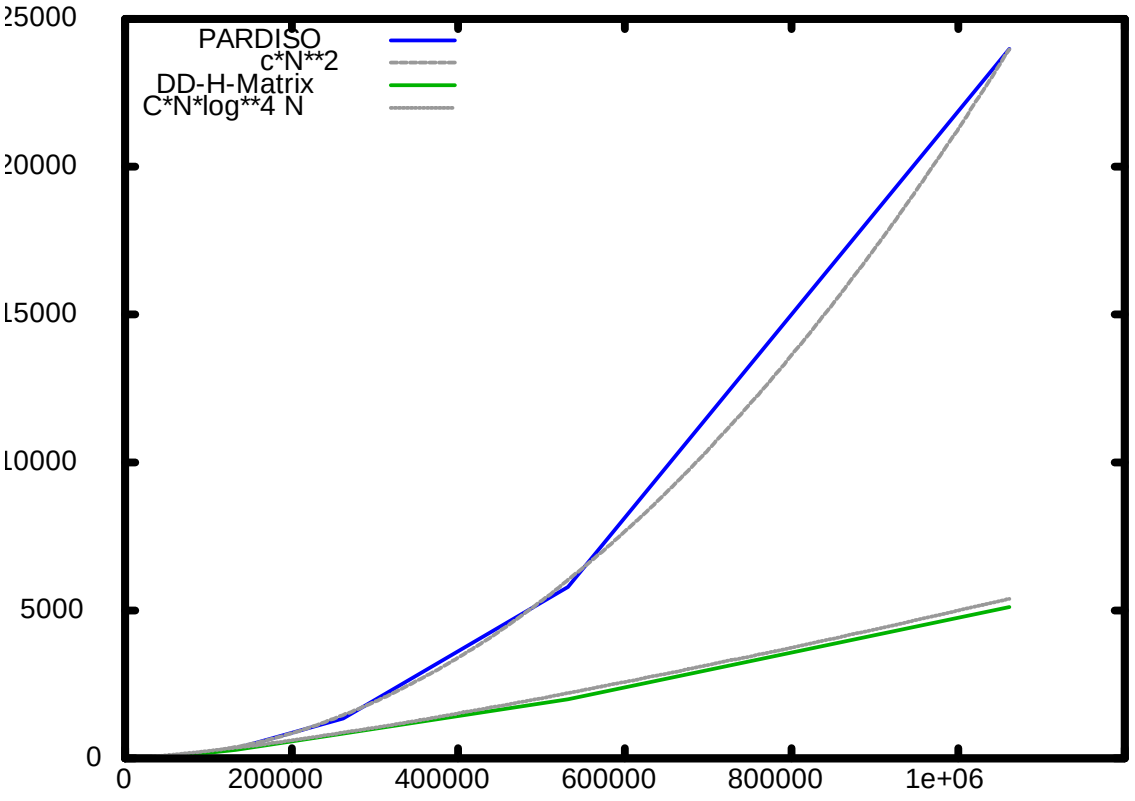
†Pardiso (direct solver by Schenk & Co)

‡PEBBLES (algebraic multigrid code by Langer/Haase)

Comparisons




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


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

5 Matrix Equations

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

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 multi-grid methods.

Matrix-Riccati Equation

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

LEMMA: The solution X satisfies

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

where

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

LEMMA: 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 sign(S).

Example of a matrix-Riccati equation: $A = \Delta_h$ (1D)

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

	$n = 101$	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, computation by Dr. L. Grasedyck

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

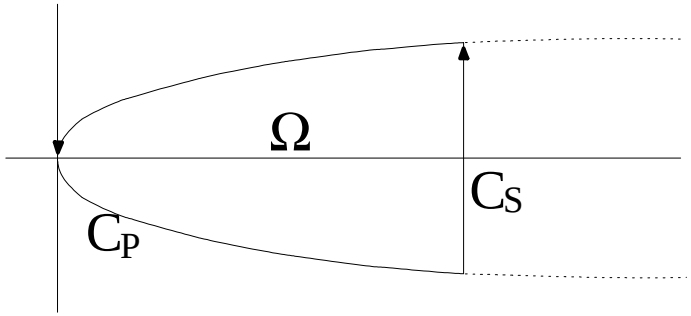
6 Matrix-Valued Functions $f(A)$

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

Cauchy-Dunford representation

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

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 $t \geq t_0 > 0$:

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

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

7 Beyond Hierarchical Matrices: Tensor Systems as Higher-dimensional Analogue

Tensor space $\mathbb{V} := V_1 \otimes V_2 \otimes \ldots \otimes V_d$.

DEFINITION: A rank-k-tensor is of the form

$$\sum_{\mu=1}^{k} v_1^{(\mu)} \otimes v_2^{(\mu)} \otimes \ldots \otimes v_d^{(\mu)} \qquad \text{with } v_j^{(\mu)} \in V_j.$$

QUESTION: Given $v \in \mathbb{V}$, are there rank-k-approximations $\tilde{v}$?
How can they be computed?

$V_i = \mathbb{R}^{n_i \times m_i} \Rightarrow \otimes$ denotes the Kronecker product of matrices.

QUESTION: Given $M = \sum_{\mu=1}^{k_M} M_1^{(\mu)} \otimes M_2^{(\mu)} \otimes \ldots \otimes M_d^{(\mu)}$. Under what conditions can the eigenvectors be approximated by rank-k-tensors?

Example from the electronic Schrödinger equation

Hartree-Fock equation $F_\psi\,\psi_b(\mathbf{y}) = \epsilon_b\,\psi_b(\mathbf{y})$ involves the Hartree potential

$$V_H(\mathbf{x}) = 2\sum_{b=1}^{N/2}\int \frac{\psi_b^*(\mathbf{y})\psi_b(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}d\mathbf{y} = \int \frac{\rho(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,d\mathbf{y}, \tag{1}$$

where $\rho(\mathbf{y}) = 2\sum_{b=1}^{N/2}\psi_b(\mathbf{y})\psi_b^*(\mathbf{y})$ is the electron density.

Standard approaches use Gaussians $g_k^{(j)}(y_j) = (y_j - A_k^{(j)})^{\ell_k}\,e^{-\alpha_k(y_j - A_k^{(j)})^2}$ to represent the orbital (wavefunction) by

$$\psi_b(\mathbf{y}) \approx \sum_{k=1}^{K_\psi} g_k^{(1)}(y_1)\,g_k^{(2)}(y_2)\,g_k^{(3)}(y_3). \tag{2}$$

Here, K_ψ = tensor rank. We start with a representation (2) produced by the MOLPRO program package using the MATROP program for matrix operations. Eq. (2) yields $\rho(\mathbf{y}) = \psi_b^*(\mathbf{y})\psi_b(\mathbf{y})$ with $K := K_\psi(K_\psi+1)/2$ terms.

Optimising the tensor representation reduces the tensor rank to a much smaller rank κ while almost keeping the same order of accuracy:

$$\rho(\mathbf{y}) \approx \sum_{k=1}^{\kappa} \varrho_k^{(1)}(y_1)\, \varrho_k^{(2)}(y_2)\, \varrho_k^{(3)}(y_3), \qquad \kappa \ll K.$$

The computational work for evaluating the Hartree potential (1) depends essentially on the tensor rank.

EXAMPLE CH_4: The MOLPRO program yields $K = 1540$, which can be reduced by our approach to $\kappa = 45$. The computing time for evaluating V_H for the tensor representation with $\kappa = 45$ is 8 hours, while the estimated time for $K = 1540$ is 190 hours.

molecule	initial rank $K$of $\rho(y)$	final rank κ	relative error	error in energy (hartree)
CH_4	1540	45	9.0×10^{-6}	6.0×10^{-5}
C_2H_2	2346	50	1.3×10^{-4}	5.0×10^{-4}
C_2H_6	4656	55	8.8×10^{-5}	4.0×10^{-4}

Separable PDE in $[0,1]^d$, d large

Let $\Omega = (0,1)^d \subset \mathbb{R}^d$; equidistant grid: $\Omega_h = (h, 2h, \ldots, nh)$ with

$$(n+1)\,h = 1.$$

Here: $n = 1024$.

Separable PDE: $L = \sum_{\nu=1}^{d} a_\nu(x_\nu)\frac{\partial^2}{\partial x_\nu^2}$, e.g., $L = \Delta$.

Discretisation of $-L$ by usual difference formula:

$$A = -L_h = -\sum_{\nu=1}^{d} a_\nu(x_\nu) D^h_{x_\nu x_\nu} \qquad (D^h_{x_\nu x_\nu}\text{: 2nd difference})$$
$$= A_1 \times I \times \ldots \times I + I \times A_2 \times \ldots \times I + \ldots + I \times I \times \ldots \times A_d.$$

Goal: Approximation of L_h^{-1}.

Numerical result (Grasedyck 2004):
For $d = 2048$, accuracy 10^{-5} to 10^{-6}: 5 min computer time

Related dimension:

$$N = 1024^{2048} = 1.24 \times 10^{6165}.$$

Underlying method

$1/x$ can be approximated by exponential sums $\sum_{\nu=1}^{k} \omega_\nu \exp(\alpha_\nu x)$:

$$\min_{\omega_\nu,\alpha_\nu} \max_{x\in[x_0,x_1]} \left|\frac{1}{x} - \sum_{\nu=1}^{k} \omega_\nu \exp(\alpha_\nu x)\right| \leq O(e^{-ck}), \quad c > 0,$$

$$\min_{\omega_\nu,\alpha_\nu} \max_{x\in[x_0,\infty)} \left|\frac{1}{x} - \sum_{\nu=1}^{k} \omega_\nu \exp(\alpha_\nu x)\right| \leq O(e^{-ck^{1/2}}), \quad c > 0.$$

Let $[x_0, x_1]$ or $[x_0, \infty)$ contain the spectrum of L_h. Then

$$L_h^{-1} \approx \sum_{\nu=1}^{k} \omega_\nu \exp(\alpha_\nu L_h).$$

The special tensor structure

$$L_h = \sum_{\mu=1}^{d} I \times \ldots \times I \times L_{h,\mu} \times I \times \ldots \times I$$

implies

$$\exp(\alpha_\nu L_h) = \bigotimes_{\mu=1}^{d} \exp(\alpha_\nu L_{h,\mu}).$$

Approximation of $\exp(\alpha_\nu L_{h,\mu})$ by $\mathcal{H}$-matrices (see above).

8 Literature etc.

W.H.: A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. Computing, 62:89–108, 1999 (Report 98-27, Oct. 1998, Universität Kiel)

W.H., B.Khoromskij, S.A.Sauter: On H^2-matrices. In Lectures on applied mathematics, S. 9–29. Springer-Verlag, Berlin, 2000 (München, June 1999)

W.H., B.Khoromskij: A sparse H-matrix arithmetic. Part II: Application to multi-dimensional problems. Computing, 64:21–47, 2000

I.P.Gavrilyuk, W.H., B.Khoromskij: H-matrix approximation for the operator exponential with applications. Numer. Math., 92:83–111, 2002.

W.H., S.Börm: Data-sparse approximation by adaptive H^2-matrices. Computing, 69:1–35, 2002.

W.H., L.Grasedyck, S.Börm: An introduction to hierarchical matrices. Math. Bohem., 127:229–241, 2002

M.Bebendorf, W.H.: Existence of H-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. Numer. Math., 95:1–28, 2003

L.Grasedyck, W.H., B.Khoromskij: Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. Computing, 70:121–165, 2003

L.Grasedyck, W.H.: Construction and arithmetics of H-matrices. Computing 70:295–334, 2003

L.Grasedyck, W.H., Sabine Le Borne. Adaptive geometrically balanced clustering of H-matrices. Computing, 73:1–23, 2004.

I.P.Gavrilyuk, W.H., B.Khoromskij: Hierarchical tensor-product approximation to the inverse and related operators for high-dimensional elliptic problems. Computing 74:131–157, 2005

W.H., B.Khoromskij, E.E.Tyrtyshnikov: Hierarchical Kronecker tensor-product approximations. J. Numer. Math. 13:119–156, 2005.

W.H., B.Khoromskij, and R.Kriemann: Direct Schur complement method by domain decomposition based on H-matrix approximation. Comput. Vis. Sci., 8:179–188, 2005.

W.H., B.Khoromskij: Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. Part I. Separable approximation of multi-variate functions. Computing, 76:177-202, 2006 - Part II. HKT representation of certain operators. Computing 76 (2006) 203-225.

L. Banjai, W.H: H- and H^2-matrices for low and high frequency Helmholtz equation. To appear in IMA J. Numer. Anal.

For a complete list see

`http://www.mis.mpg.de` (→institute reports) or

`http://www.mis.mpg.de/scicomp/hackbusch_e.html`

For scientific purposes a software library is freely available (licence form to be signed),

for commercial applications: HLibPro