H2Lib - A short introduction

...and some experiments

Nicholas Krämer

May 17, 2018

Outline

Where can one find which information?

What are $\mathcal{H}$-matrices?

Getting started with the H2Lib

The (probably) quickest way to create an $\mathcal{H}$-matrix

Some issues arising in scattered data approximation

Where can one find which information?

- The H2Lib including a **detailed** documentation:

  `http://www.h2lib.org/`

- A little bit more about hierarchical matrices:

  Lecture notes of the winter school 2003,

  Google $\rightarrow$ "hierarchical matrices lecture notes"

- More detailed: Surveys, books (Hackbusch, Bebendorf)

What does the H2Lib include?

Among others:

- Data structures for $\mathcal{H}$-, $\mathcal{H}^2$- and low rank matrices
- Functions for corresponding operations: multiplication, **compression**, factorisation, ...
- Some applications such as BEM (popular!) and FEM

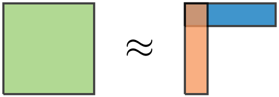
License:

- Open source (LGPL 3.0)

What are hierarchical matrices ($\mathcal{H}$-matrices)?

- Store (kernel-based) matrices in $\mathcal{O}(N \log N)$ instead of N^2
- Clustering of points $\rightarrow$ tree of pointsets
- **Tree** of full and low rank matrices
- Low rank: **Approximate** kernel using degenerate kernel, e.g.

$$g(x, y) \approx \sum_{i=1}^{k} L_i(x) g(\xi_i, y) \quad \Leftrightarrow$$



- Low rank approximations "very good" if corresponding clusters of x and y are "far apart"

What does an $\mathcal{H}$-matrix look like?

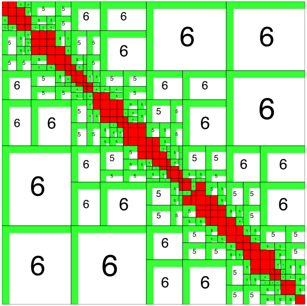


Figure: $\mathcal{H}$-matrix of size $N = 250$ in dimension $d = 1$

“the more white spots one can see the better”

Is it true that it only requires $\mathcal{O}(N \log N)$ storage?

Example 1/2: Storage of a kernelmatrix in 2D

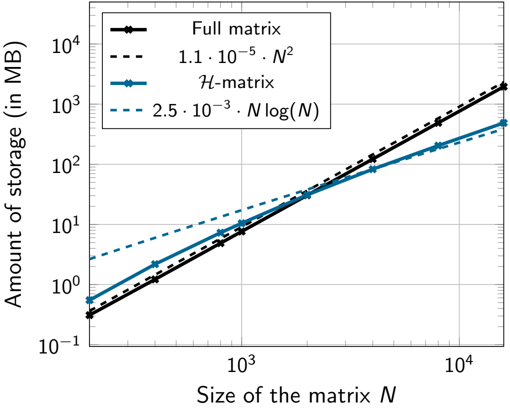


Figure: Amound of storage for a kernel matrix $K = (\Phi(\|x_i - x_j\|))_{i,j=1}^N$, depending on N for fixed precision $\epsilon = 10^{-13}$

How important is the precision?

Example 2/2: How important is the precision?

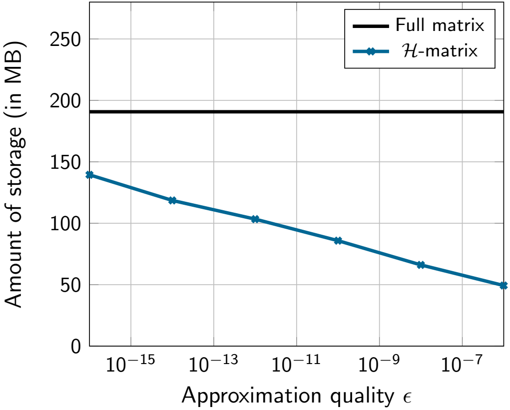


Figure: Amound of storage for a kernel matrix depending on ϵ for fixed matrix size $N = 5.000$, compression such that $\|K_H - K\|_2 / \|K\|_2 \leq \epsilon$

Getting started with the H2Lib

Installation

1. Download: `http://www.h2lib.org/`
2. Make use of the makefile:

```
cd Documents/H2Lib/
make
```

Where are the modules?

- In the header files, for example `Library/hmatrix.h`
- The library itself is contained in `libh2.a` (has to be linked):

```
gcc test.c -L/home/kraemer/Documents/H2Lib/ -lh2
```

What is the quickest way to initialize and fill an $\mathcal{H}$-matrix using the H2Lib?

Sketched procedure for initializing an $\mathcal{H}$-matrix

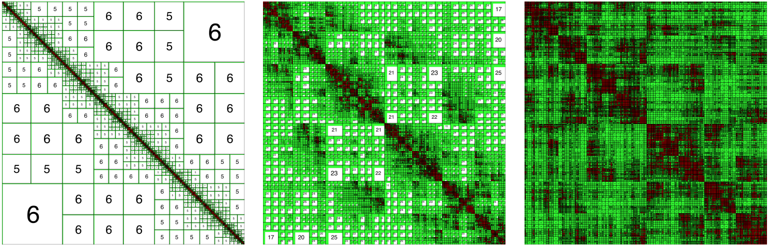
1. Create clustergeometry and fill it with points
```
geom = new_clustergeometry(dim,N)
```
2. Create **two** clustertrees using the geometry and index set
```
{rc,cc} = build_cluster(geom,N,idx,leaf,type)
```
3. Combine clustertrees to create a block tree
```
bk = build_strict_block(rc,cc,adpar, adcond)
```
4. Initialize the $\mathcal{H}$-matrix and fill it with values
```
H = build_from_block_hmatrix(bk,localrank),
```

Sketched procedure for $\mathcal{H}$-matrices in BEM

1. Pick
   - Quadrature points for (reg. and sing.) integrals
   - Basis functions (const./linear "directly available")
   - Grid(points) (cubes, spheres, etc. "directly available")
2. BEM object: SLP & DLP (Laplace-/Helmholtz "dir. av.")
3. Cluster tree, block tree, $\mathcal{H}$-matrix

   $\rightarrow$ via `build_bem2d_*`, `setup_bem2d_*`, etc.
4. Solvers: e.g. Krylov (CG, ...), factorizations (LU, ...),...

Some issues arising in scattered data approximation (and perhaps elsewhere)



(a) $d = 1$ (b) $d = 2$ (c) $d = 3$

Figure: $\mathcal{H}$-matrix compressions of a kernel matrix of size $N = 2000$ using the IMQ kernel and scattered data $X \subseteq [0, 1]^2$ in different dimensions

Problem 2/2: Solving a (very ill-conditioned) system

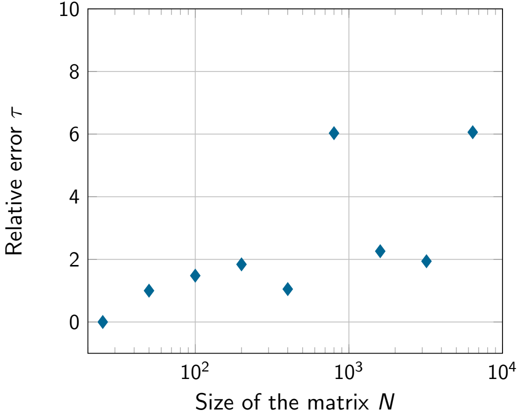


Figure: Relative error $\tau = \frac{\|x^* - x_H\|_2}{\|x^*\|_2}$ of the solutions obtained via LU decomposition for the $\mathcal{H}$-matrix and full matrix in dimension $d = 1$

$\mathcal{H}(^2)$-matrices and Gaussian processes

Some problems look like: solve $(K + \sigma I)x = y$ for certain $\sigma > 0$

N	$\sigma = 10^{-15}$	$\sigma = 10^{-8}$	$\sigma = 10^{-1}$
25	0.1	$2.3 \cdot 10^{-8}$	$4.8 \cdot 10^{-15}$
100	0.1	$1.9 \cdot 10^{-7}$	$2.2 \cdot 10^{-14}$
400	14.7	$6.8 \cdot 10^{-7}$	$1.0 \cdot 10^{-13}$
1600	1.2	$1.4 \cdot 10^{-6}$	$2.2 \cdot 10^{-13}$
6400	3.3	$3.1 \cdot 10^{-6}$	$6.6 \cdot 10^{-13}$

Possible extensions:

- Randomized linear algebra ([ASKIT; Yu, Biros, March; 2017])
- Fast matrix-vector-multiplication (+Krylov solvers), on-the-fly evaluation of kernels ([Garcke, Börm; 2007])

Summary

1. The H2Lib is well-documented and easy to find
2. Hierarchical matrices are trees of matrices
3. They approximate
4. They indeed have $\mathcal{O}(N \log N)$ storage
5. Clustergeometry $\rightarrow$ cluster tree $\rightarrow$ block tree $\rightarrow$ $\mathcal{H}$-matrix
6. Not perfect for very ill-conditioned systems
7. Good for BEM and $(K + \sigma I)x = y$

Some texts I might have mentioned

ASKIT:
"An $N \log N$ parallel fast direct solver for kernel matrices"; C. Yu, W. March, G. Biros; 2017

$\mathcal{H}^2$-arithmetics and kernel matrices:
"Approximating Gaussian processes with $\mathcal{H}^2$-matrices"; S. Börm and J. Garcke; 2007

Hierarchical matrices in scattered data approximation:
"Hierarchical matrix approximation for kernel-based scattered data interpolation"; A. Iske, S. Le Borne, M. Wende; 2017