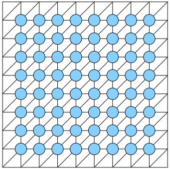
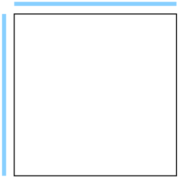
Alternative Clustering for Sparse Matrices

Clustering

Domain:

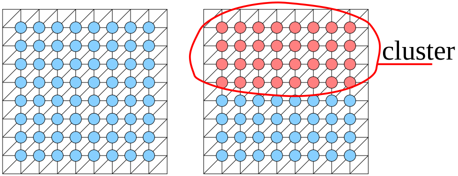


Matrix:

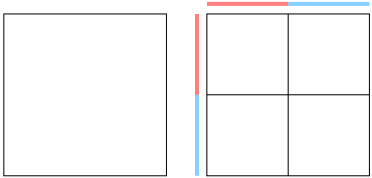


Clustering

Domain:

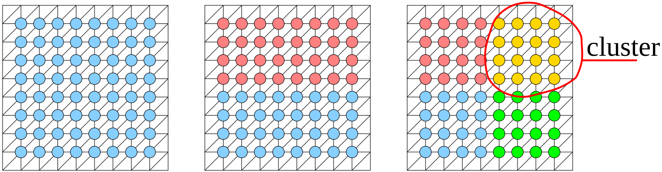


Matrix:

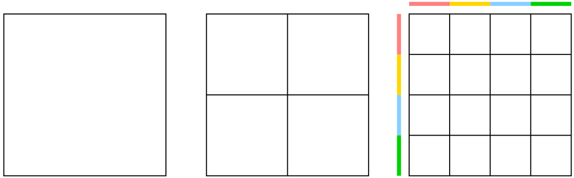


Clustering

Domain:

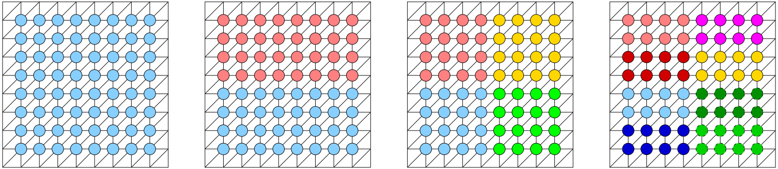


Matrix:

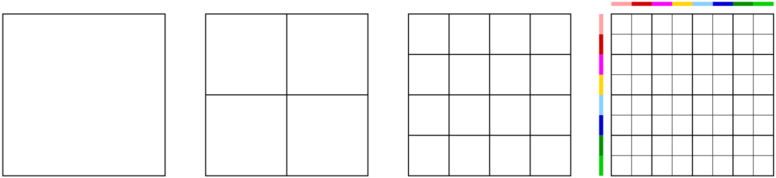


CLUSTERING

Domain:

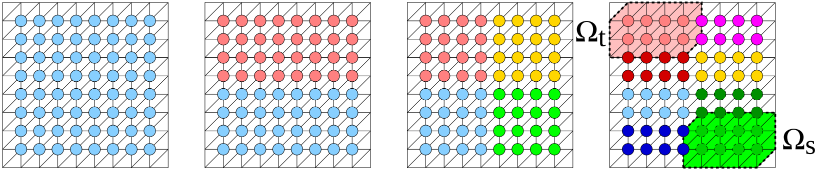


Matrix:

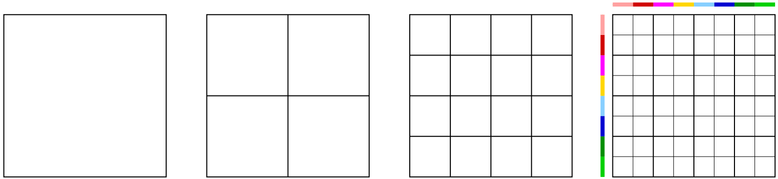


CLUSTERING

Domain:

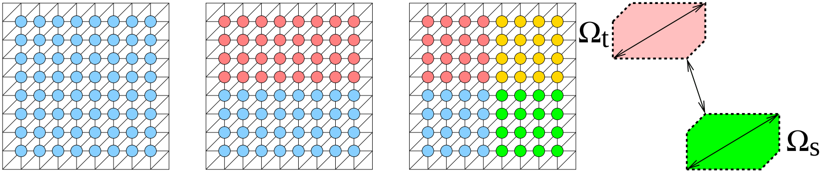


Matrix:

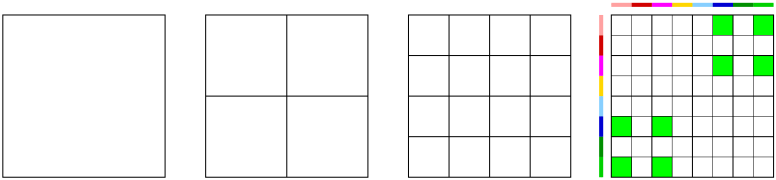


CLUSTERING

Domain:

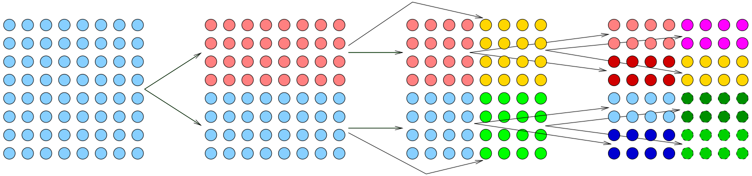


Matrix:

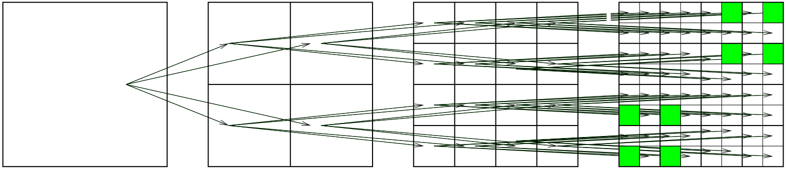


CLUSTERING

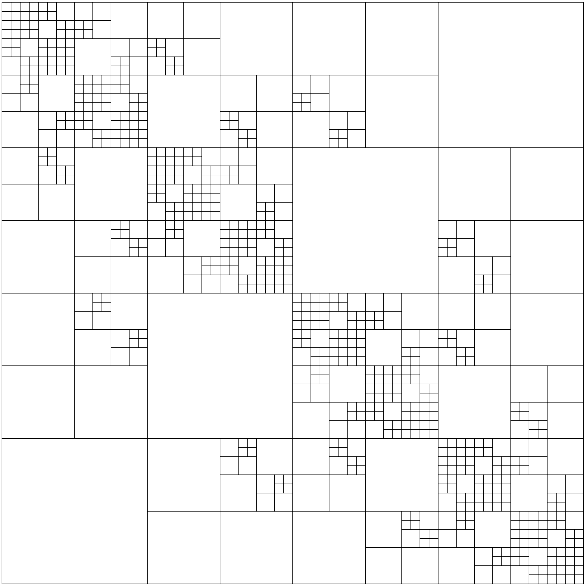
Domain:



Matrix:

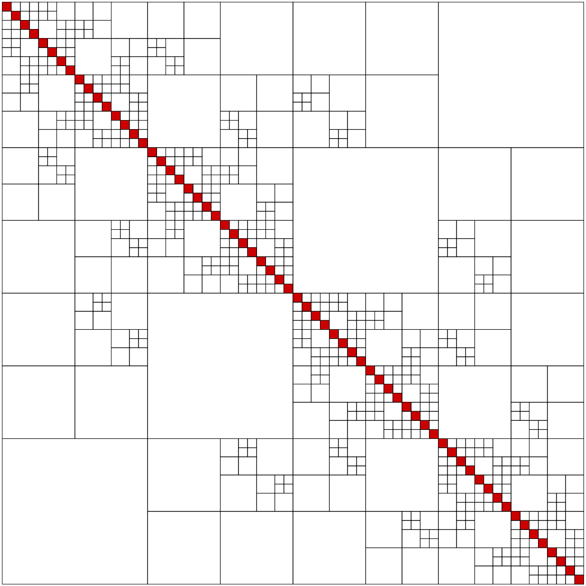


MATRIX STRUCTURE



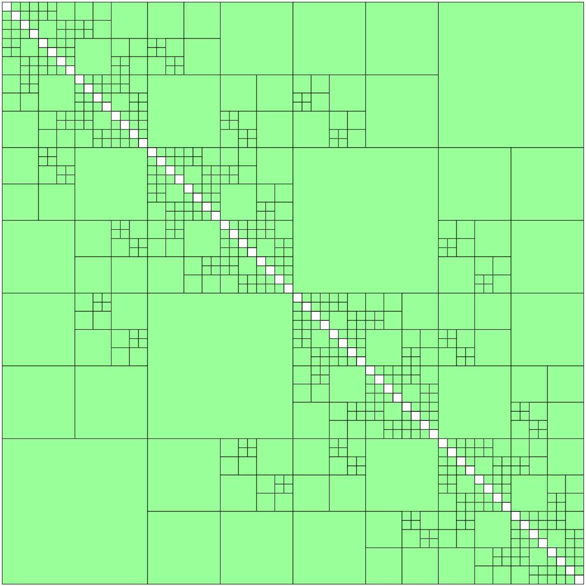
- $\mathcal{O}(n)$ blocks

Matrix Structure



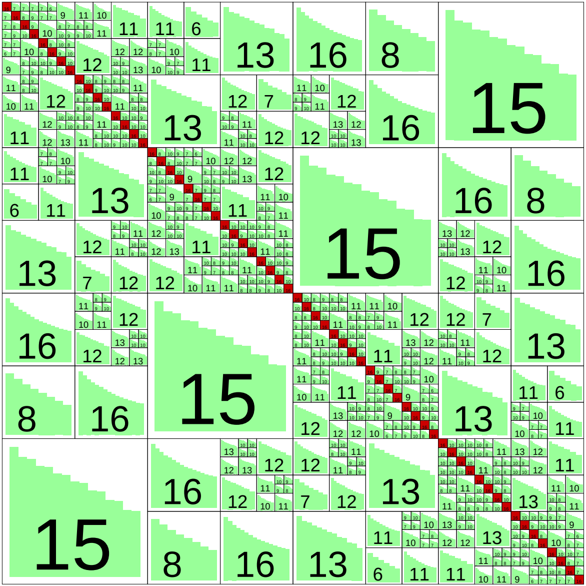
- $\mathcal{O}(n)$ blocks
- Small red blocks: `fullmatrix`

Matrix Structure



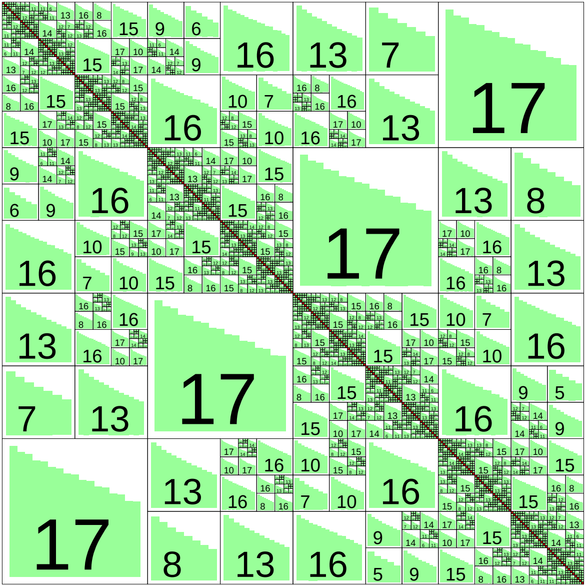
- $\mathcal{O}(n)$ blocks
- Small red blocks: `fullmatrix`
- All other blocks: `rkmatrix`

Matrix Structure



- blockwise: exponential decay of singular values
- matrix size: 1024×1024 3.2 MB

MATRIX STRUCTURE

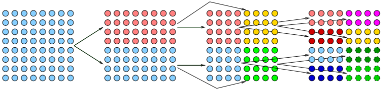


- blockwise: exponential decay of singular values
- matrix size: 4096×4096 20.7 MB

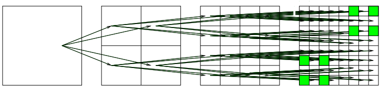
$\mathcal{H}$-Matrix Arithmetic

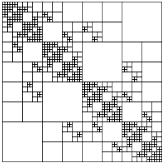
How do we calculate with $\mathcal{H}$-matrices ?

1. Construct **cluster tree** T_I
   → geometry information
   → black box (`sparsematrix` graph)



2. Construct **block cluster tree** $T_{I\times I}$
   → partition of matrix
   → requires admissibility condition

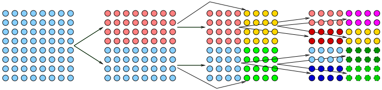




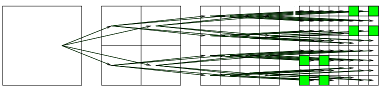
$\mathcal{H}$-Matrix Arithmetic

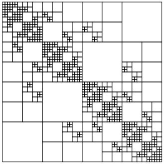
How do we calculate with $\mathcal{H}$-matrices ?

1. Construct **cluster tree** T_I
   → geometry information
   → black box (`sparsematrix` graph)



2. Construct **block cluster tree** $T_{I \times I}$
   → partition of matrix
   → requires admissibility condition



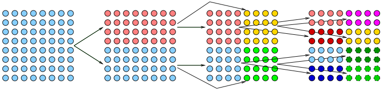


3. Convert `sparsematrix` A into $\mathcal{H}$-Matrix H

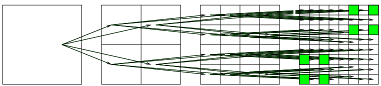
$\mathcal{H}$-Matrix Arithmetic

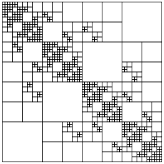
How do we calculate with $\mathcal{H}$-matrices ?

1. Construct **cluster tree** T_I
   $\rightarrow$ geometry information
   $\rightarrow$ black box (`sparsematrix` graph)



2. Construct **block cluster tree** $T_{I\times I}$
   $\rightarrow$ partition of matrix
   $\rightarrow$ requires admissibility condition



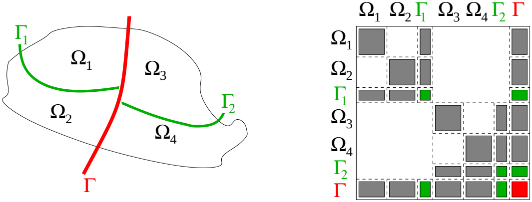


3. Convert `sparsematrix` A into $\mathcal{H}$-Matrix H
4. Compute $\mathcal{H}$**-LU** factorization

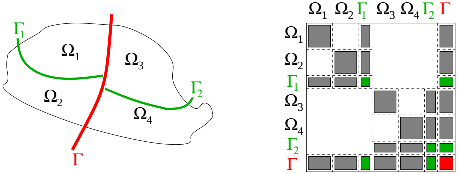
Result:

$$\mathbf{A} \approx \mathbf{L}\,\mathbf{U}, \qquad L, U \in \mathcal{H}(T_{I\times I})$$

Domain Decomposition & Nested Dissection



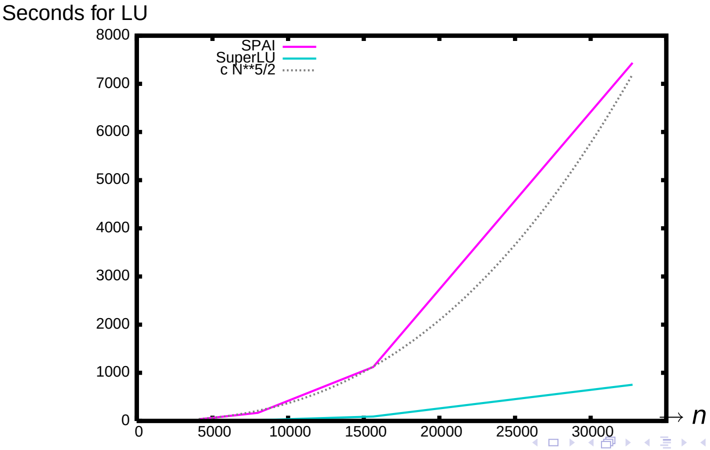
Nested dissection and matrix structure



matrix structure of LU-factorization

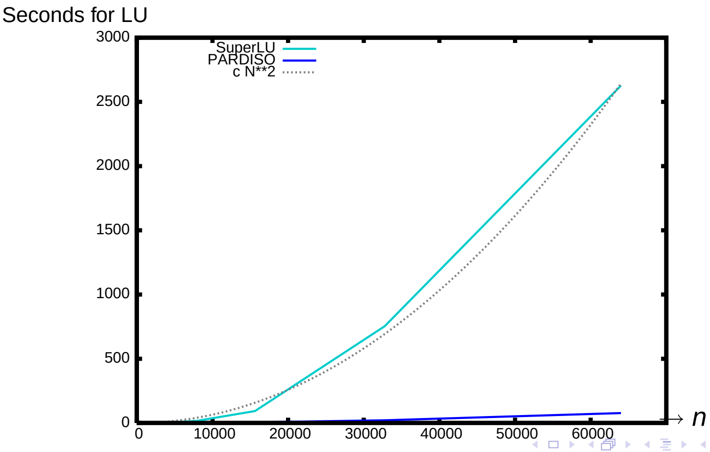
Domain Decomposition & Nested Dissection

- SPAI (Barnard/Grote/Broeker/Hagemann)
- SuperLU (Demmel/Gilbert/Li)



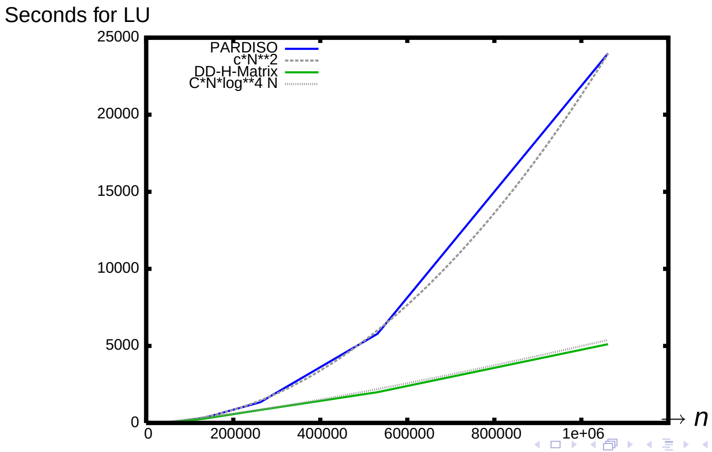
Domain Decomposition & Nested Dissection

- SuperLU (Demmel/Gilbert/Li)
- PARDISO (Schenk/Gärtner/Fichtner)



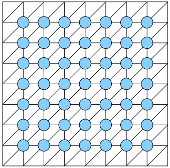
DOMAIN DECOMPOSITION & NESTED DISSECTION

- PARDISO (Schenk/Gärtner/Fichtner)
- DD-H-Matrix (Grasedyck/Kriemann/LeBorne)

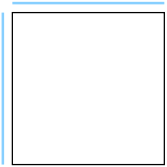


Domain Decomposition & Nested Dissection

Domain:

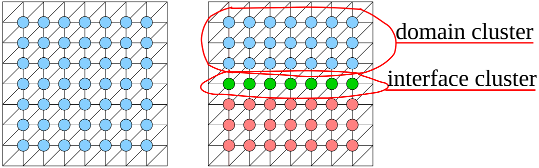


Matrix:

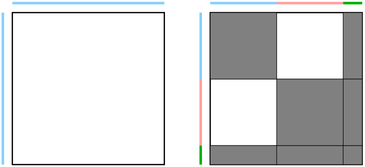


Domain Decomposition & Nested Dissection

Domain:

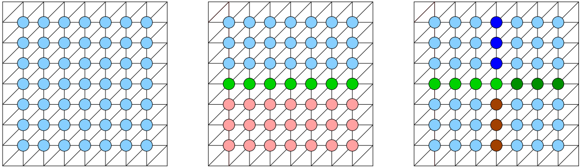


Matrix:

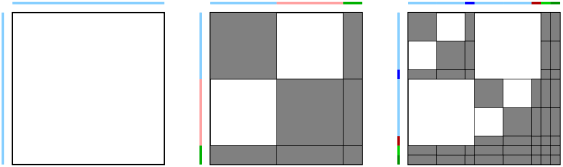


DOMAIN DECOMPOSITION & NESTED DISSECTION

Domain:

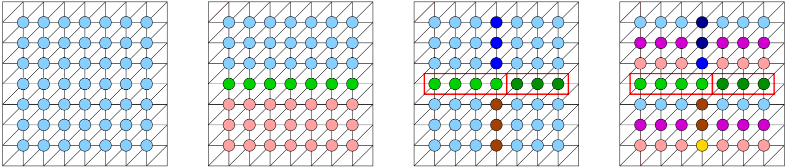


Matrix:

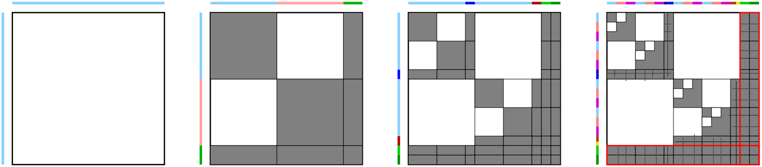


Domain Decomposition & Nested Dissection

Domain:

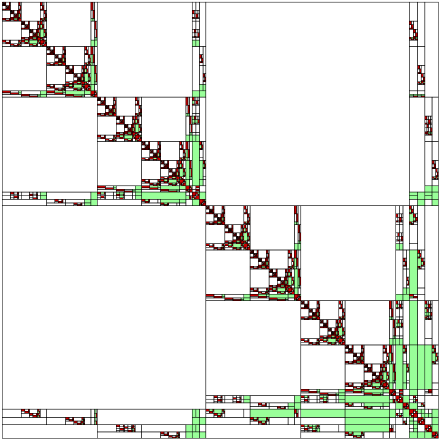


Matrix:

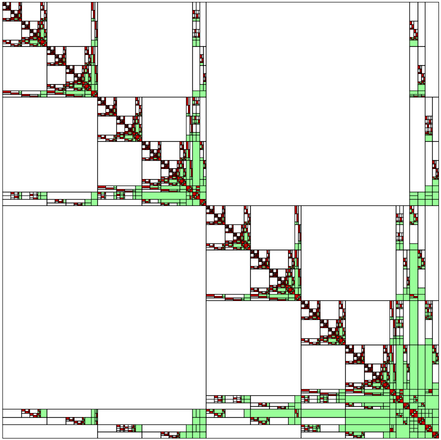


Domain Decomposition & Nested Dissection

Example ($d = 3$)

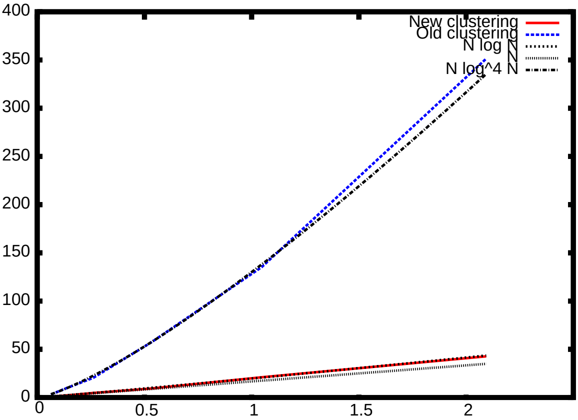


Matrix A.



$\mathcal{H}$-LU decomposition

OLD AND NEW CLUSTERING



$\mathcal{H}$-LU versus DD-$\mathcal{H}$-LU for 2d-Laplace

$\mathcal{H}^2$-MATRICES

$\mathcal{H}$-Matrices:

1. Storage: $\mathcal{O}(N \log(N) k)$
2. LU: $\mathcal{O}(N \log^2(N) k^2)$
3. Rank $k \sim \log^{d-2}(1/\varepsilon)$
4. Total $\mathcal{O}(N \log^{2d-2}(N))$

$\mathcal{H}^2$-Matrices

$\mathcal{H}$-Matrices:

1. Storage: $\mathcal{O}(N \log(N) k)$
2. LU: $\mathcal{O}(N \log^2(N) k^2)$
3. Rank $k \sim \log^{d-2}(1/\varepsilon)$
4. Total $\mathcal{O}(N \log^{2d-2}(N))$

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

1. Storage: $\mathcal{O}(Nk)$
2. LU: $\mathcal{O}(N \log^2(N) k^2)$
3. Rank $k \sim \log^{d-1}(1/\varepsilon)$
4. Total $\mathcal{O}(N \log^{2d}(N))$

$\mathcal{H}^2$-Matrices

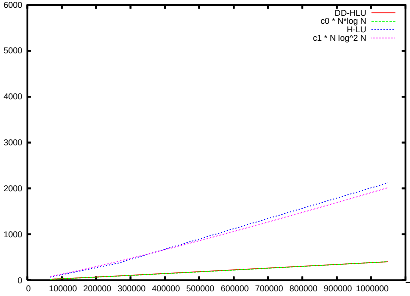
$\mathcal{H}$-Matrices:

1. Storage: $\mathcal{O}(N \log(N) k)$
2. LU: $\mathcal{O}(N \log^2(N) k^2)$
3. Rank $k \sim \log^{d-2}(1/\varepsilon)$
4. Total $\mathcal{O}(N \log^{2d-2}(N))$

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

1. Storage: $\mathcal{O}(Nk)$
2. LU: $\mathcal{O}(N \log^2(N) k^2)$
3. Rank $k \sim \log^{d-1}(1/\varepsilon)$
4. Total $\mathcal{O}(N \log^{2d}(N))$

Seconds for LU



$\longrightarrow N$