Formats of Structured Matrices

And the Quest for Memory Minimization

Ronald Kriemann
Max Planck Institute MiS Leipzig

siam 2024 Conference on Applied Linear Algebra

MAX PLANCK INSTITUTE
FOR MATHEMATICS IN THE SCIENCES

Hierarchical Matrices

Hierarchical Matrices

Approximate dense data $M_{\tau,\sigma} \in \mathbb{C}^{\#\tau \times \#\sigma}$ of $M \in \mathbb{C}^{n \times n}$ by

$$U_{\tau,\sigma} \cdot V_{\tau,\sigma}^H$$

with $U_{\tau,\sigma} \in \mathbb{C}^{\#\tau \times k}$, $V_{\tau,\sigma} \in \mathbb{C}^{\#\sigma \times k}$ and

$$k \ll \min(\#\tau, \#\sigma)$$

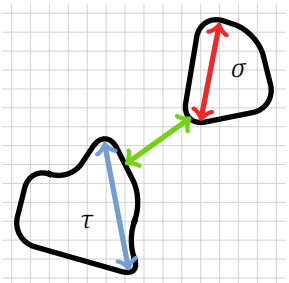
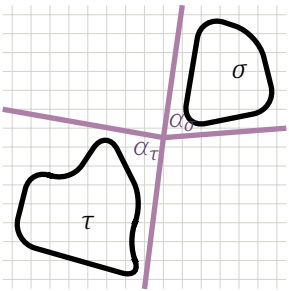
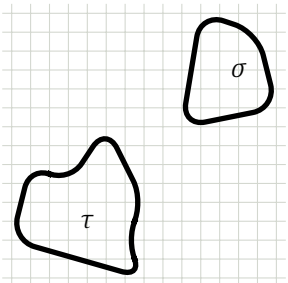
such that

$$||M_{\tau,\sigma} - U_{\tau,\sigma} V_{\tau,\sigma}^H|| \leq \delta \qquad \text{or}$$
$$||M_{\tau,\sigma} - U_{\tau,\sigma} V_{\tau,\sigma}^H|| \leq \varepsilon ||M_{\tau,\sigma}||$$

yielding an approximation $\widetilde{M}$ of M with $\mathcal{O}(n \log^{\alpha} n)$ storage.

In the literature, many different formats of (hierarchical) lowrank matrices exist.

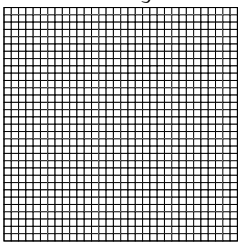
Admissibility

Strong	Weak[1]	Off-Diagonal
		
$\min(\operatorname{diam}(\tau), \operatorname{diam}(\sigma)) \leq \eta \operatorname{dist}(\tau, \sigma)$	$\max(\alpha_\tau, \alpha_\sigma) \leq \alpha$	$\tau \neq \sigma$

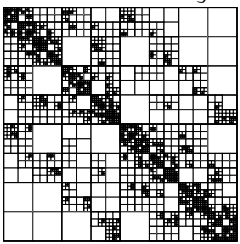
[1] Hackbusch, Khoromskij, K.: "*Hierarchical Matrices Based on a Weak Admissibility Criterion*", Computing 73, 2004

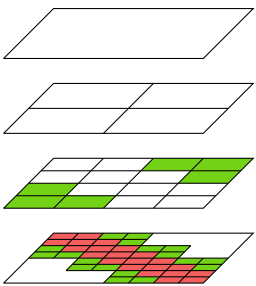
Hierarchy

Flat Layout



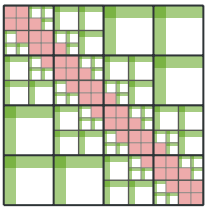
Full Hierarchy





Basis Representation

Separate

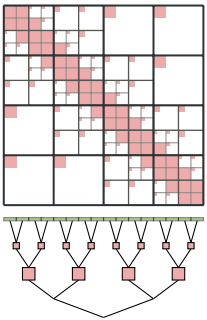


$$M_{\tau,\sigma} = U_{\tau,\sigma} \cdot V_{\tau,\sigma}^H$$

with

$$U_{\tau,\sigma} \in \mathbb{R}^{\#\tau \times k},\ V_{\tau,\sigma} \in \mathbb{R}^{\#\sigma \times k}$$

Nested

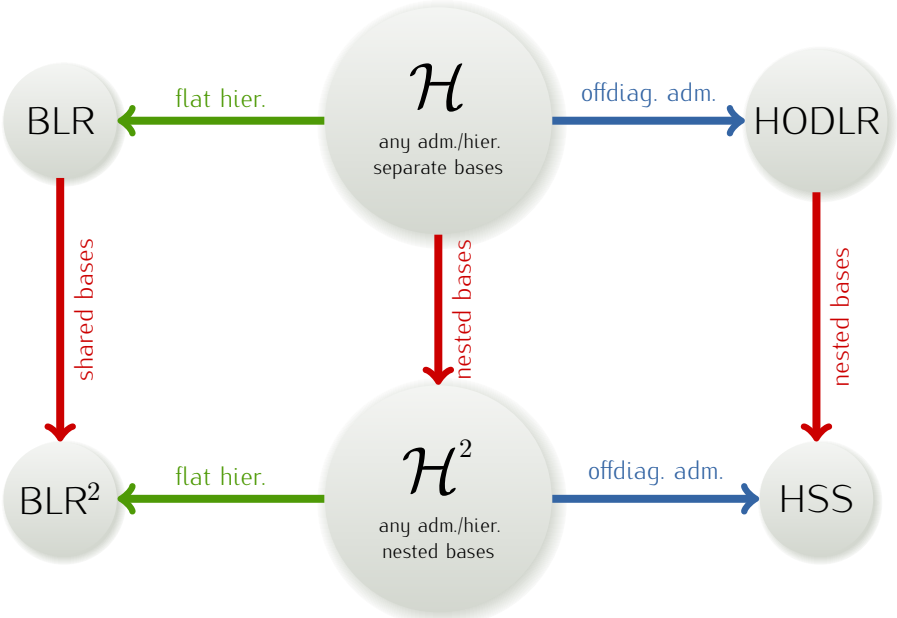


$$M_{\tau,\sigma} = \widetilde{\mathcal{U}}_\tau \cdot S_{\tau,\sigma} \cdot \widetilde{\mathcal{V}}_\sigma^H$$

with

implicit $\widetilde{\mathcal{U}}_\tau, \widetilde{\mathcal{V}}_\sigma$

Formats of (Rank-) Structured Matrices



Optimized Low-Rank Storage

Floating Point Compression

Idea

Replace FP64 (or FP32) for lowrank/dense storage by a *lossy* compression scheme.

Floating Point Compression

Idea

Replace FP64 (or FP32) for lowrank/dense storage by a *lossy* compression scheme.

Options

- compression libraries: ZFP[1,2], BLOSC, ... (*not* SZ/SZ3, MGARD)

[1] K., Ltaief, Luong, Pérez, Im, Keyes: "*High-Performance Spatial Data Compression for Scientific Applications*", Euro-Par 2022

[2] Claus, Ghysels, Liu, Nhan, Thirumalaisamy, Bhalla, Li: "*Sparse Approximate Multifrontal Factorization with Composite Compression Methods*", ACM Trans. Math. Softw., 2023

Floating Point Compression

Idea

Replace FP64 (or FP32) for lowrank/dense storage by a *lossy* compression scheme.

Options

- compression libraries: ZFP[1,2], BLOSC, ... (*not* SZ/SZ3, MGARD)
- IEEE-754 with adaptive mantissa and exponent bits,

[1] K., Ltaief, Luong, Pérez, Im, Keyes: "*High-Performance Spatial Data Compression for Scientific Applications*", Euro-Par 2022

[2] Claus, Ghysels, Liu, Nhan, Thirumalaisamy, Bhalla, Li: "*Sparse Approximate Multifrontal Factorization with Composite Compression Methods*", ACM Trans. Math. Softw., 2023

Floating Point Compression

Idea

Replace FP64 (or FP32) for lowrank/dense storage by a *lossy* compression scheme.

Options

- compression libraries: ZFP[1,2], BLOSC, ... (*not* SZ/SZ3, MGARD)
- IEEE-754 with adaptive mantissa and exponent bits,
- Posits,
- ...

[1] K., Ltaief, Luong, Pérez, Im, Keyes: "*High-Performance Spatial Data Compression for Scientific Applications*", Euro-Par 2022

[2] Claus, Ghysels, Liu, Nhan, Thirumalaisamy, Bhalla, Li: "*Sparse Approximate Multifrontal Factorization with Composite Compression Methods*", ACM Trans. Math. Softw., 2023

Floating Point Compression

Idea

Replace FP64 (or FP32) for lowrank/dense storage by a *lossy* compression scheme.

Options

- compression libraries: ZFP, BLOSC, ... (*not* SZ/SZ3, MGARD)
- IEEE-754 with adaptive mantissa and exponent bits,
- Posits,
- ...

Used

Based on IEEE-754 scheme. Choose

- mantissa bits: $m := \lceil -\log_2 \varepsilon \rceil$,
- exponent bits: $e := \lceil \log_2 \log_2 v_{\max}/v_{\min} \rceil$

Round up m for byte-aligned, fast storage.

Adaptive Precision for Low-Rank Data

Given $||M_{\tau,\sigma} - U_{\tau,\sigma}V_{\tau,\sigma}^H|| \leq \delta$ and p floating point formats and

$$U_{\tau,\sigma}V_{\tau,\sigma}^H = W\Sigma X^H = \begin{pmatrix} W_1 \dots W_p \end{pmatrix} \begin{pmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_p \end{pmatrix} \begin{pmatrix} X_1 \dots X_p \end{pmatrix}^H$$

with unit roundoffs $u_1, \dots, u_p$ such that

$$||\Sigma_i|| \leq \frac{\delta}{u_i}$$

Let $\widetilde{M}_{\tau,\sigma}$ be the representation of $M_{\tau,\sigma}$ where W_i, X_i are stored in the i'th floating point format. Then the error $||M_{\tau,\sigma} - \widetilde{M}_{\tau,\sigma}||$ is bounded by[1]

$$||M_{\tau,\sigma} - \widetilde{M}_{\tau,\sigma}|| \leq \delta + \left(2(p-1) + \sum_{i=2}^{p} \sqrt{k_i} u_i \right) \delta$$

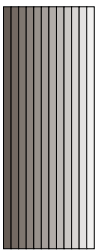
[1] Amestoy, Boiteau, Buttari, Gerest, Jézéquel, L'Excellent, Mary: *"Mixed precision low-rank approximations and their application to block low-rank LU factorization"*, IMA J. of Num. Analysis, 2022

Adaptive Precision for Low-Rank Data

Adaptive Precision for Low-Rank (APLR)

Using a given direct floating point compression scheme *choose* precision $\widetilde{u}_i$ for *each* column (w_i, x_i) of W/X such that

$$\widetilde{u}_i = \frac{\delta}{\sigma_i}$$



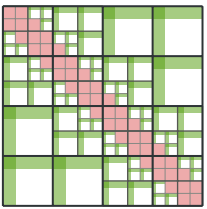
For the total approximation error one then obtains:

$$||M_{\tau,\sigma} - \widetilde{M}_{\tau,\sigma}|| \leq \delta + \left(2k\delta + \delta^2 \sum_{i=1}^{k} \frac{1}{\sigma_i} \right)$$

Adaptive Precision for Low-Rank Data

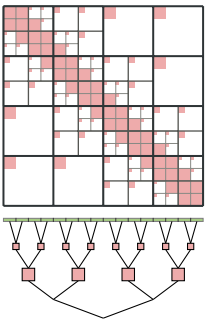
Depending on the matrix format, APLR can be applied to different data.

Separate Bases



all *lowrank blocks* ($\mathcal{O}(n \log n)$)

Nested Bases



all *leaf* cluster bases ($\mathcal{O}(n)$)

For *everything else* direct compression is applied!

Results

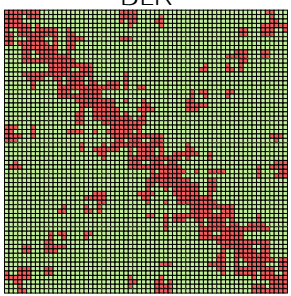
Laplace SLP

$$\int_\Omega \frac{1}{|x-y|} u(x)dy = f(x), \quad x \in \Omega$$

with $\Omega = \left\{x \in \mathbb{R}^3 : |x|_2 = 1\right\}$.

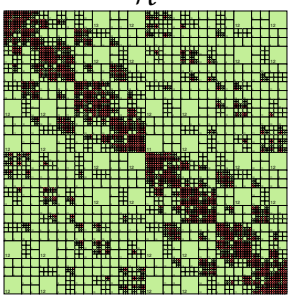
Discretization with piecewise constant ansatz functions.

BLR



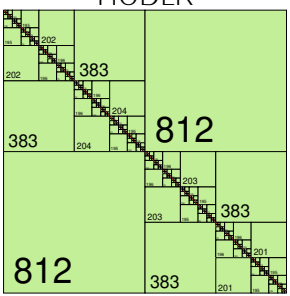
$6 \leq k \leq 14$

$\mathcal{H}$



$6 \leq k \leq 15$

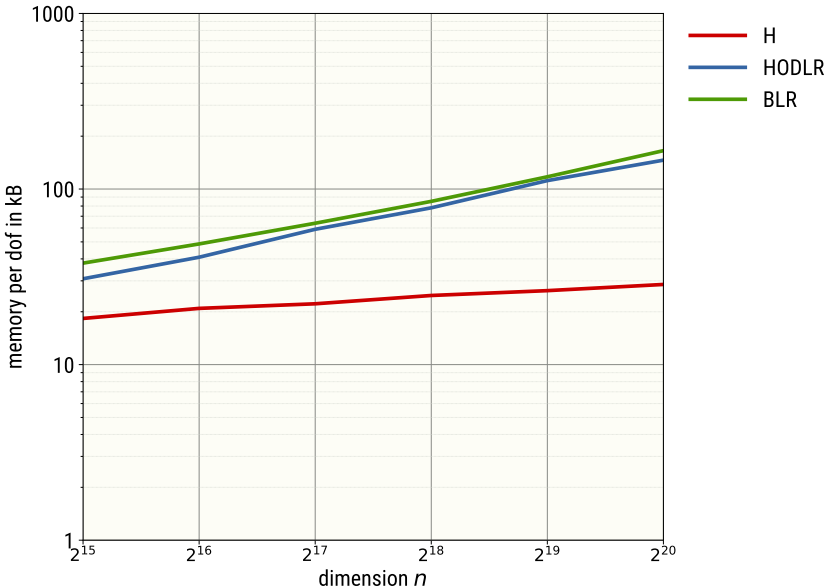
HODLR



$25 \leq k \leq 812$

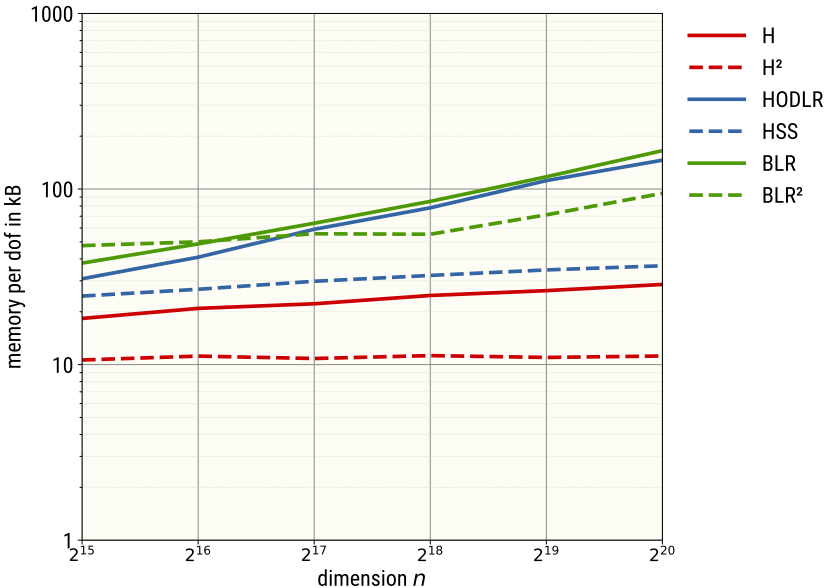
Results

Laplace SLP ($\varepsilon = 10^{-6}$)



Results

Laplace SLP ($\varepsilon = 10^{-6}$)



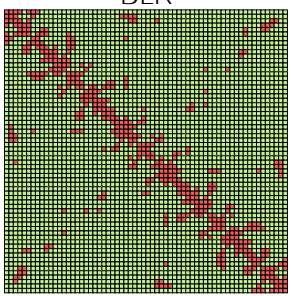
Results

Matérn Covariance

$$C(d_{ij}, \sigma, \ell, \nu) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} d_{ij} \right)^{\nu} \mathcal{K}_\nu \left(\frac{\sqrt{2\nu}}{\ell} d_{ij} \right),$$

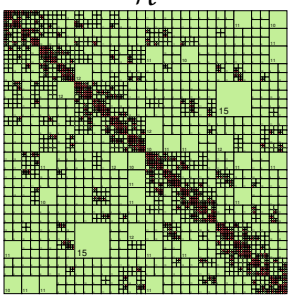
with distance $d_{ij} = |x_i - x_j|_2$ between points[1] $x_i, x_j \in \left\{ x \in \mathbb{R}^3 : |x|_2 \leq 1 \right\}$.

BLR



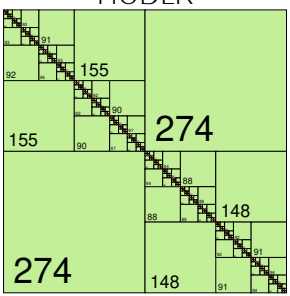
$5 \leq k \leq 17$

$\mathcal{H}$



$6 \leq k \leq 19$

HODLR

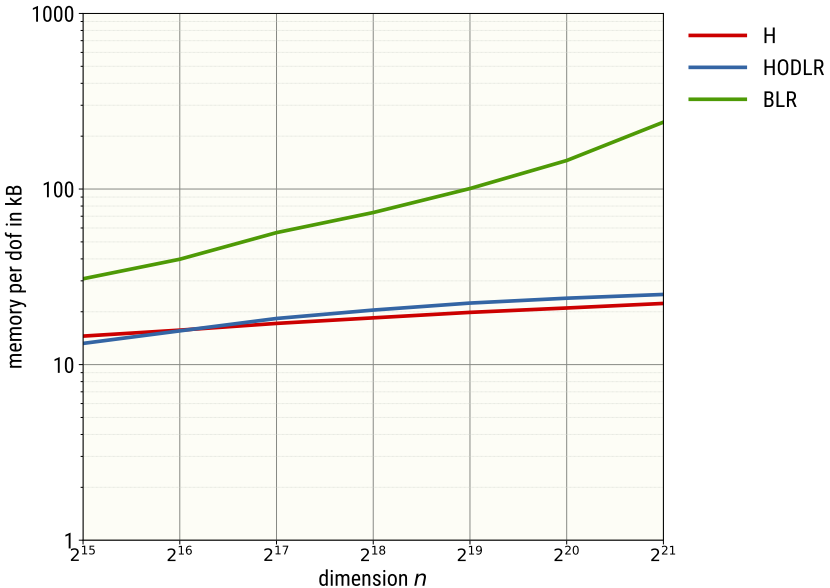


$16 \leq k \leq 274$

[1] Randomly chosen, fixed seed.

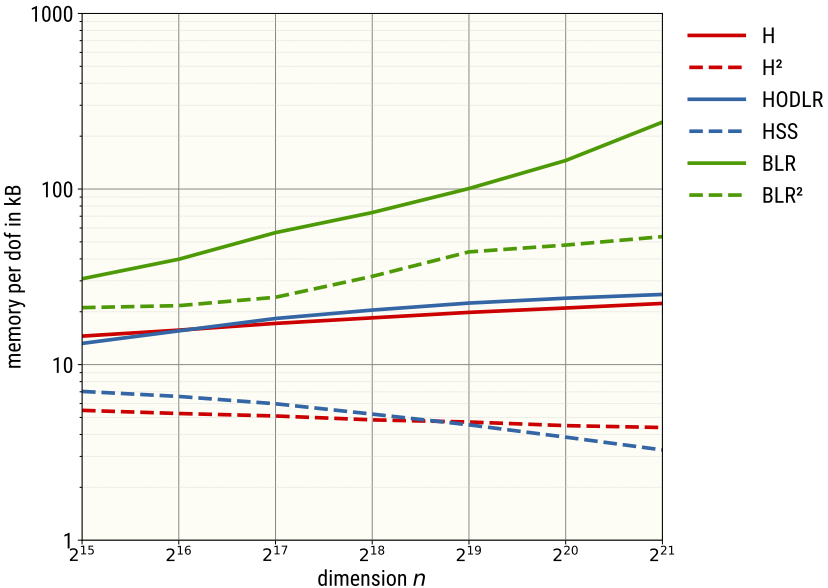
Results

Matérn Covariance ($\varepsilon = 10^{-6}$)



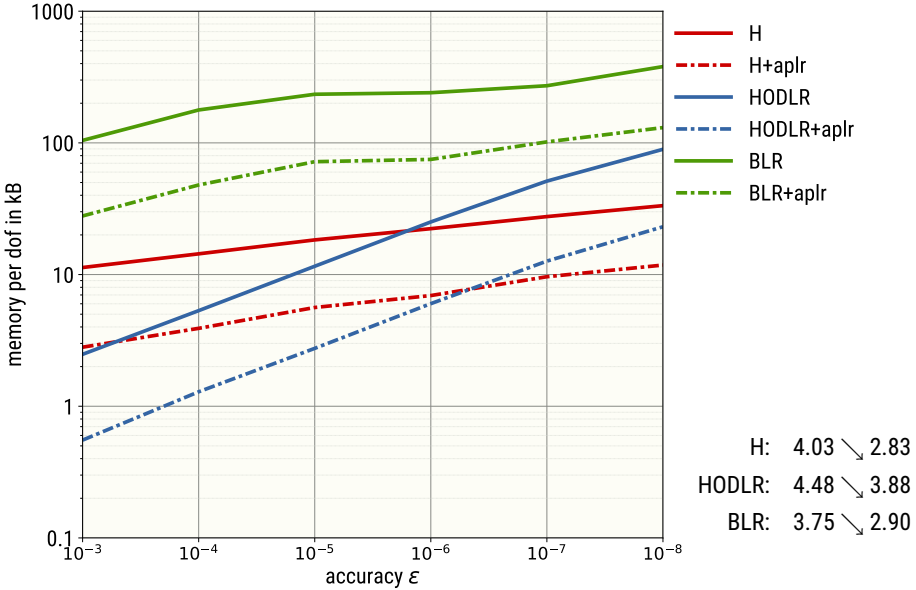
Results

Matérn Covariance ($\varepsilon = 10^{-6}$)



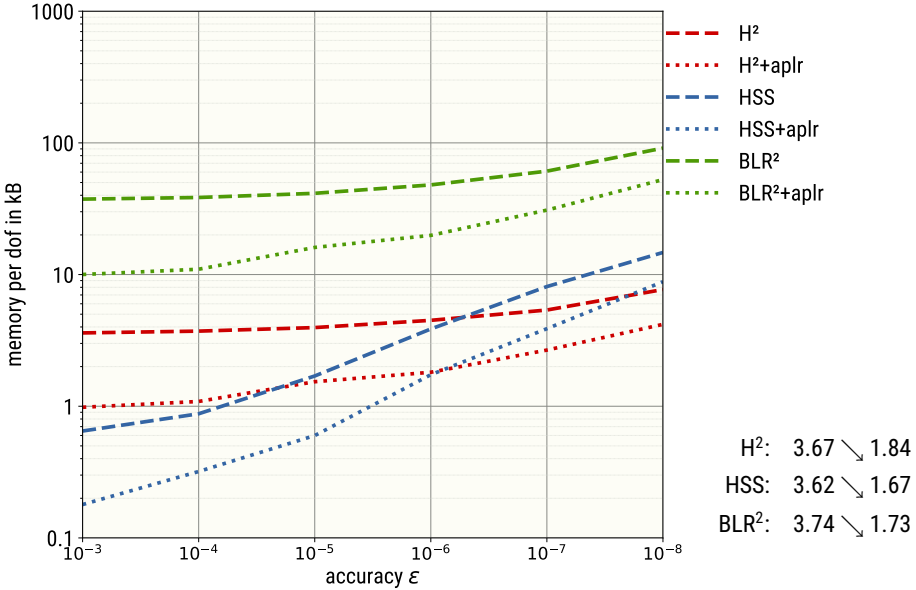
Results

Matérn Covariance ($n = 1.048.576$)



Results

Matérn Covariance ($n = 1.048.576$)

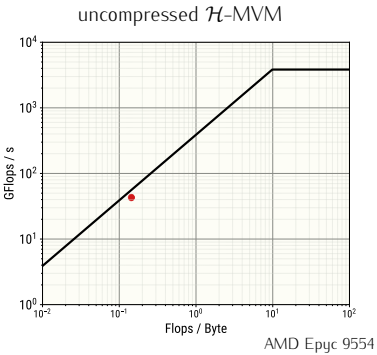


$\mathcal{H}$-Matrix-Vector Multiplication

$\mathcal{H}$-Matrix-Vector Multiplication

$\mathcal{H}$-MatVec is *bandwidth* limited on most systems.

No arithmetic performance improvements possible.

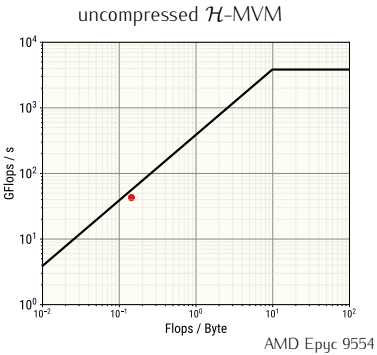


[1] Anzt, Flegar, Grützmacher, Quintana-Ortí: "*Toward a modular precision ecosystem for high-performance computing*", Int. J. of HPC Applications, 33(6), 1069–1078, 2019.

$\mathcal{H}$-Matrix-Vector Multiplication

$\mathcal{H}$-MatVec is *bandwidth* limited on most systems.

No arithmetic performance improvements possible.



Compressed $\mathcal{H}$-MatVec

Decoupling of storage and compute precision[1]:

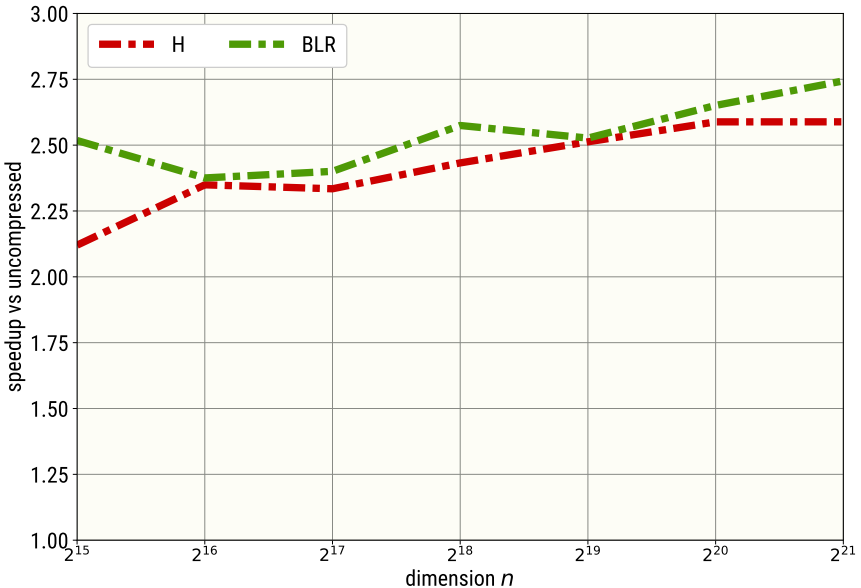
- compression only for storage,
- all computations in FP64.
- *on-the-fly* decompression

```
function MVM(in: U, V, x, inout: y)
  t := 0;
  for 0 ≤ ℓ < k do
    for 0 ≤ j < m do
      t_ℓ := t_ℓ + decompress(V_jℓ) x_j;
  for 0 ≤ ℓ < k do
    for 0 ≤ i < n do
      y_i := y_i + decompress(U_iℓ) t_ℓ;
```

[1] Anzt, Flegar, Grützmacher, Quintana-Ortí: "*Toward a modular precision ecosystem for high-performance computing*", Int. J. of HPC Applications, 33(6), 1069–1078, 2019.

$\mathcal{H}$-Matrix-Vector Multiplication

Matérn Covariance ($\varepsilon = 10^{-6}$)

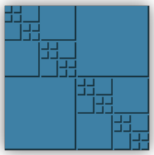


Conclusion

Conclusion

With APLR for (hierarchical) lowrank matrices

- storage demands are *significantly reduced*
- and performance of matrix-vector multiplication *much improved*.



libHLR.org

arxiv.org/abs/2308.10960

arxiv.org/abs/2405.03456

Thank You

MAX PLANCK INSTITUTE
FOR MATHEMATICS IN THE SCIENCES