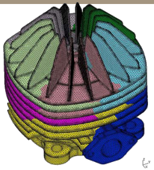
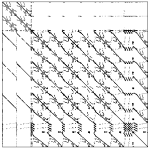
Bridging the gap between flat and hierarchical low-rank matrix formats

P. Amestoy[1] A. Buttari[2] J.-Y. L'Excellent[3] T. Mary[4]

[1]INP-IRIT [2]CNRS-IRIT [3]INRIA-LIP [4]University of Manchester

Structured Matrix Days, Lyon, 14-15 May 2018

 $\rightarrow$  $\rightarrow$ 

Linear system $Ax = b$

Often a keystone in scientific computing applications
(discretization of PDEs, step of an optimization method, ...)

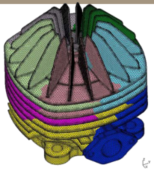
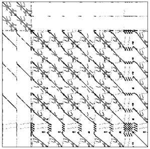
Direct methods

Factorize $A = LU$ and solve $LUx = b$

- ☺ Numerically reliable
- ☹ Computational cost

Theo Mary (contact: theo.mary@manchester.ac.uk)

Context

 $\rightarrow$  $\rightarrow$ 

Linear system $Ax = b$

Often a keystone in scientific computing applications
(discretization of PDEs, step of an optimization method, ...)

Direct methods

Factorize $A = LU$ and solve $LUx = b$

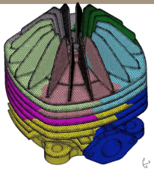
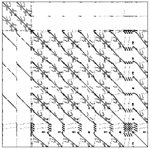
☺ Numerically reliable

☹ Computational cost

Objective:
reduce the cost of direct methods ...
...while maintaining their numerical reliability

Theo Mary (contact: theo.mary@manchester.ac.uk)

Context

 $\rightarrow$  $\rightarrow$ 

Large scale applications

- Target size is $n \sim 10^9$ for sparse $\Rightarrow$ $m \sim 10^6$ for dense
- $O(m^2)$ storage complexity and $O(m^3)$ flop complexity
  $m \sim 10^6 \Rightarrow$ TeraBytes of storage and ExaFlops of computation!

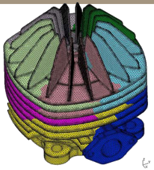
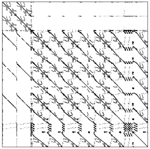
$\Rightarrow$ Need to reduce the asymptotic complexity

Large scale systems

- Increasingly large numbers of cores available

$\Rightarrow$ Need to design parallel algorithms

Theo Mary (contact: theo.mary@manchester.ac.uk)

 $\rightarrow$  $\rightarrow$ 

Large scale applications

- Target size is $n \sim 10^9$ for sparse $\Rightarrow$ $m \sim 10^6$ for dense
- $O(m^2)$ storage complexity and $O(m^3)$ flop complexity
  $m \sim 10^6 \Rightarrow$ TeraBytes of storage and ExaFlops of computation!

$\Rightarrow$ Need to reduce the asymptotic complexity

Large scale systems

- Increasingly large numbers of cores available

$\Rightarrow$ Need to design parallel algorithms

These two objectives are not always compatible

Theo Mary (contact: theo.mary@manchester.ac.uk)

Outline

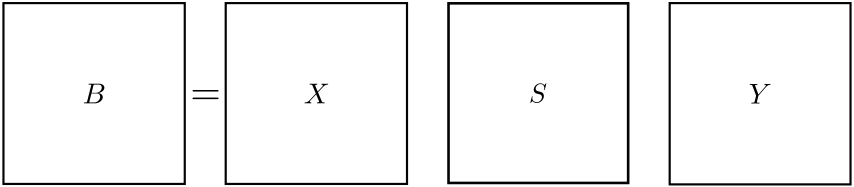
1. Introduction
   - The $\mathcal{H}$ format: very good complexity
   - The BLR format: very good parallelism
2. Motivation
   - Why we need a new format to bridge the gap
3. The MBLR format
   - Complexity analysis
   - Numerical results
4. Conclusion

Preprint

P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary, *Bridging the gap between flat and hierarchical low-rank matrix formats: the multilevel BLR format*, submitted (2018).

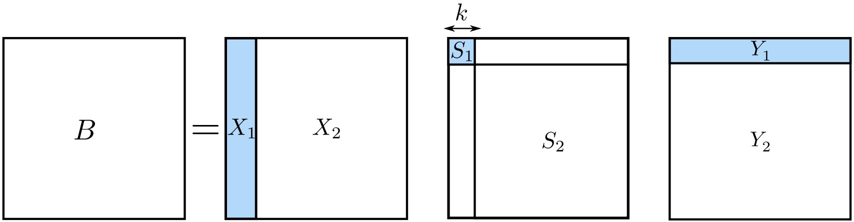
Introduction

Take a dense matrix B of size $b \times b$ and compute its SVD $B = XSY$:



Theo Mary (contact: theo.mary@manchester.ac.uk)

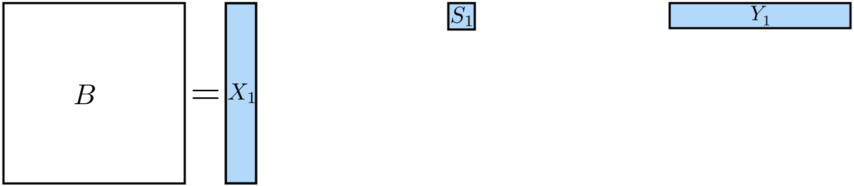
Take a dense matrix B of size $b \times b$ and compute its SVD $B = XSY$:



$k = \min \{k \leq n; \sigma_{k+1} \leq \varepsilon\}$ is the numerical rank at accuracy ε

Theo Mary (contact: theo.mary@manchester.ac.uk)

Take a dense matrix B of size $b \times b$ and compute its SVD $B = XSY$:

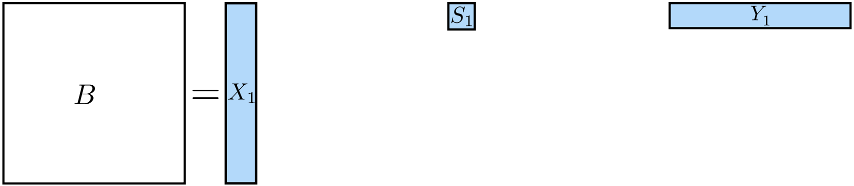


$k = \min\{k \le n; \sigma_{k+1} \le \varepsilon\}$ is the numerical rank at accuracy ε

$\tilde{B} = X_1 S_1 Y_1$ is a low-rank approximation to B: $\|B - \tilde{B}\|_2 \le \varepsilon$

Low-rank matrices

Take a dense matrix B of size $b \times b$ and compute its SVD $B = XSY$:



$k = \min\{k \le n; \sigma_{k+1} \le \varepsilon\}$ is the numerical rank at accuracy ε

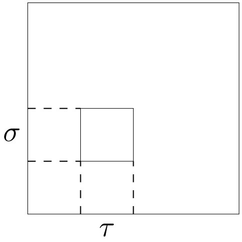
$\tilde{B} = X_1 S_1 Y_1$ is a low-rank approximation to B: $\|B - \tilde{B}\|_2 \le \varepsilon$

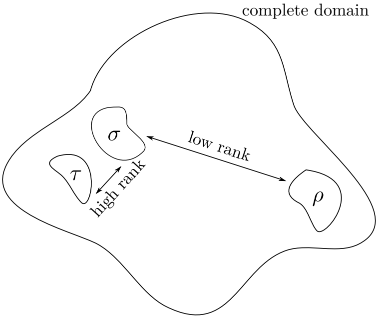
Storage savings: $b^2/2bk = b/2k$
Similar flops savings when used in most linear algebra kernels

 Theo Mary (contact: theo.mary@manchester.ac.uk)

Low-rank blocks

Most matrices are not low-rank in general but in some applications they exhibit low-rank blocks



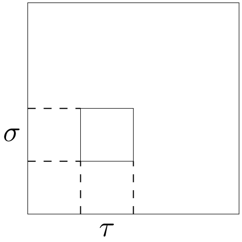


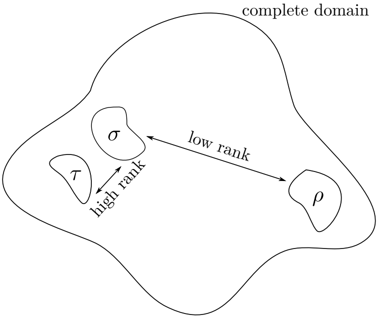
A block B represents the interaction between two subdomains σ and τ.
Small diameter and far away $\Rightarrow$ low numerical rank.

Theo Mary (contact: theo.mary@manchester.ac.uk)

Low-rank blocks

Most matrices are not low-rank in general but in some applications they exhibit low-rank blocks

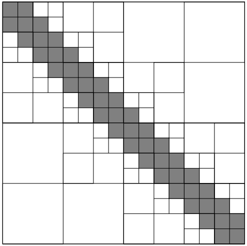




A block B represents the interaction between two subdomains σ and τ.
Small diameter and far away $\Rightarrow$ low numerical rank.

How to choose a good block partitioning of the matrix?

Theo Mary (contact: theo.mary@manchester.ac.uk)

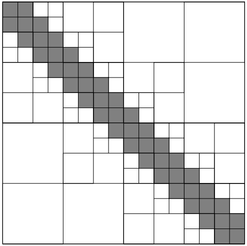


$\mathcal{H}$-matrix

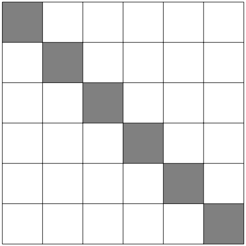
- Nearly linear complexity
- Complex, hierarchical structure

Theo Mary (contact: theo.mary@manchester.ac.uk)

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$-matrix

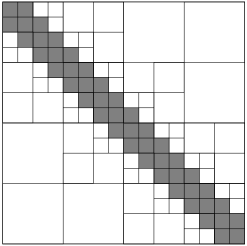


BLR matrix

- Nearly linear complexity
- Complex, hierarchical structure

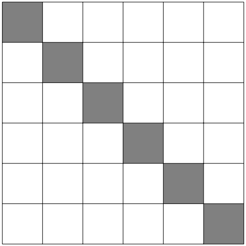
- Superlinear complexity
- Simple, flat structure

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$-matrix

- Nearly linear complexity
- Complex, hierarchical structure



BLR matrix

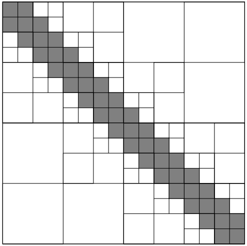
- Superlinear complexity
- Simple, flat structure

BLR is a comprise between complexity and performance:

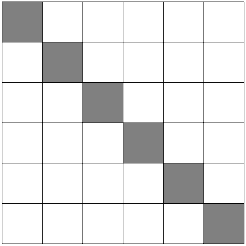
- Small blocks $\Rightarrow$ can fit on single shared-memory node
- No global order between blocks $\Rightarrow$ flexible data distribution
- Easy to handle numerical pivoting

Theo Mary (contact: theo.mary@manchester.ac.uk)

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$-matrix



BLR matrix

- Nearly linear complexity
- Complex, hierarchical structure

- Superlinear complexity
- Simple, flat structure

BLR is a comprise between complexity and performance:

- Small blocks $\Rightarrow$ can fit on single shared-memory node
- No global order between blocks $\Rightarrow$ flexible data distribution
- Easy to handle numerical pivoting

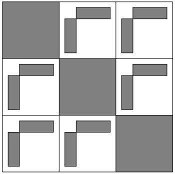
Can we find an even better comprise?

Theo Mary (contact: theo.mary@manchester.ac.uk)

Motivation

Computing the BLR complexity

Assume all off-diagonal blocks are low-rank. Then:

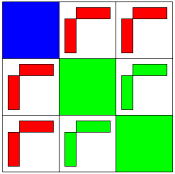


$$
\begin{aligned}
Storage &= cost_{LR} * nb_{LR} + cost_{FR} * nb_{FR} \\
&= O(br) * O((\frac{m}{b})^2) + O(b^2) * O(\frac{m}{b}) \\
&= O(m^2 r/b + mb) \\
&= \mathbf{O(m^{3/2} r^{1/2})} \text{ for } b = (mr)^{1/2}
\end{aligned}
$$

Theo Mary (contact: theo.mary@manchester.ac.uk)

Computing the BLR complexity

Assume all off-diagonal blocks are low-rank. Then:



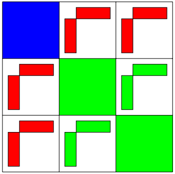
$$\begin{aligned}
Storage &= cost_{LR} * nb_{LR} + cost_{FR} * nb_{FR} \\
&= O(br) * O((\frac{m}{b})^2) + O(b^2) * O(\frac{m}{b}) \\
&= O(m^2 r/b + mb) \\
&= \mathbf{O(m^{3/2} r^{1/2})} \text{ for } b = (mr)^{1/2}
\end{aligned}$$

$$\begin{aligned}
FlopLU &= cost_{getrf} * nb_{getrf} + cost_{trsm} * nb_{trsm} + cost_{gemm} * nb_{gemm} \\
&= O(b^3) * O(\frac{m}{b}) + O(b^2 r) * O((\frac{m}{b})^2) + O(br^2) * O((\frac{m}{b})^3) \\
&= O(mb^2 + m^2 r + m^3 r^2 / b^2) \\
&= \mathbf{O(m^2 r)} \text{ for } b = (mr)^{1/2}
\end{aligned}$$

Theo Mary (contact: theo.mary@manchester.ac.uk)

Computing the BLR complexity

Assume all off-diagonal blocks are low-rank. Then:



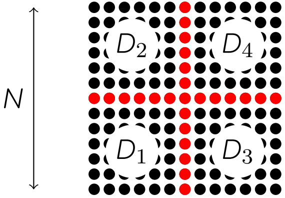
$$\begin{aligned} \textit{Storage} &= \textit{cost}_{LR} * \textit{nb}_{LR} + \textit{cost}_{FR} * \textit{nb}_{FR} \\ &= O(br) * O((\frac{m}{b})^2) + O(b^2) * O(\frac{m}{b}) \\ &= O(m^2 r/b + mb) \\ &= \mathbf{O(m^{3/2} r^{1/2})} \text{ for } b = (mr)^{1/2} \end{aligned}$$

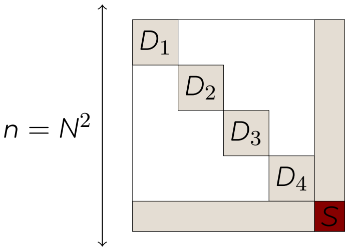
$$\begin{aligned} \textit{FlopLU} &= \textit{cost}_{getrf} * \textit{nb}_{getrf} + \textit{cost}_{trsm} * \textit{nb}_{trsm} + \textit{cost}_{gemm} * \textit{nb}_{gemm} \\ &= O(b^3) * O(\frac{m}{b}) + O(b^2 r) * O((\frac{m}{b})^2) + O(br^2) * O((\frac{m}{b})^3) \\ &= O(mb^2 + m^2 r + m^3 r^2/b^2) \\ &= \mathbf{O(m^2 r)} \text{ for } b = (mr)^{1/2} \end{aligned}$$

Result holds if a **constant** number of off-diag. blocks is full-rank.

P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary, *On the Complexity of the Block Low-Rank Multifrontal Factorization*, SIAM J. Sci. Comput. (2017).

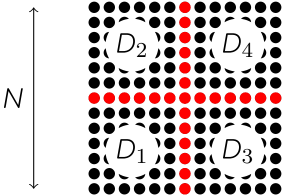
From dense to sparse: nested dissection

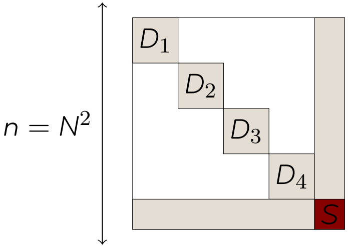


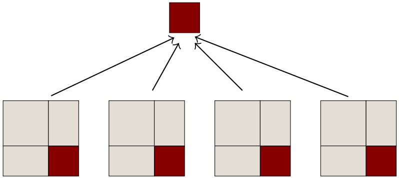


 Theo Mary (contact: theo.mary@manchester.ac.uk)

From dense to sparse: nested dissection







Proceed recursively to compute separator tree

Factorizing a sparse matrix amounts to factorizing a sequence of dense matrices

$\Rightarrow$

sparse complexity is directly derived from dense one

 Theo Mary (contact: theo.mary@manchester.ac.uk)

Nested dissection complexity formulas

2D: $\mathcal{C}_{sparse} = \sum_{\ell=0}^{\log N} 4^\ell \mathcal{C}_{dense}(\frac{N}{2^\ell})$

Nested dissection complexity formulas

2D: $$\mathcal{C}_{sparse} = \sum_{\ell=0}^{\log N} 4^{\ell} \mathcal{C}_{dense}(\frac{N}{2^{\ell}})$$

3D: $$\mathcal{C}_{sparse} = \sum_{\ell=0}^{\log N} 8^{\ell} \mathcal{C}_{dense}(\frac{N^2}{4^{\ell}})$$

Theo Mary (contact: theo.mary@manchester.ac.uk)

Nested dissection complexity formulas

2D: $\mathcal{C}_{sparse} = \sum_{\ell=0}^{\log N} 4^{\ell} \mathcal{C}_{dense}(\frac{N}{2^{\ell}})$ $\rightarrow$ common ratio $2^{2-\beta}$

3D: $\mathcal{C}_{sparse} = \sum_{\ell=0}^{\log N} 8^{\ell} \mathcal{C}_{dense}(\frac{N^2}{4^{\ell}})$ $\rightarrow$ common ratio $2^{3-2\beta}$

Assume $\mathcal{C}_{dense} = O(m^{\beta})$. Then:

2D		3D	
	$\mathcal{C}_{sparse}(n)$		$\mathcal{C}_{sparse}(n)$
$\beta > 2$	$O(n^{\beta/2})$	$\beta > 1.5$	$O(n^{2\beta/3})$
$\beta = 2$	$O(n \log n)$	$\beta = 1.5$	$O(n \log n)$
$\beta < 2$	$O(n)$	$\beta < 1.5$	$O(n)$

$\Rightarrow$ **Key motivation:** $\mathcal{C}_{dense} < O(m^2)$ **(2D) or** $O(m^{3/2})$ **(3D) is enough to get optimal sparse complexity!**

Theo Mary (contact: theo.mary@manchester.ac.uk)

Bridging the gap between flat and hierarchical formats

	Storage			Flop LU		
	$\mathcal{C}_{dense}$	$\mathcal{C}_{sparse}$ 2D	$\mathcal{C}_{sparse}$ 3D	$\mathcal{C}_{dense}$	$\mathcal{C}_{sparse}$ 2D	$\mathcal{C}_{sparse}$ 3D
FR	$O(m^2)$	$O(n \log n)$	$O(n^{4/3})$	$O(m^3)$	$O(n^{3/2})$	$O(n^2)$
BLR	$O(m^{3/2})$	$O(n)$	$O(n \log n)$	$O(m^2)$	$O(n \log n)$	$O(n^{4/3})$
$\mathcal{H}$	$O(m \log m)$	$O(n)$	$O(n)$	$O(m \log^2 m)$	$O(n)$	$O(n)$

Motivation:

- 2D flop and 3D storage complexity: can we find a simple way to improve just a little $\mathcal{C}_{dense}$?
- 3D flop complexity: still a large gap between BLR and $\mathcal{H}$

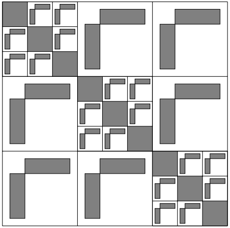
We propose a multilevel BLR format to bridge the gap

Theo Mary (contact: theo.mary@manchester.ac.uk)

The MBLR format

Complexity of the two-level BLR format

Assume all off-diagonal blocks are low-rank. Then:

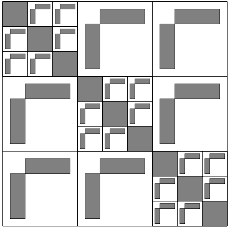


$$
\begin{aligned}
Storage &= cost_{LR} * nb_{LR} + cost_{BLR} * nb_{BLR} \\
&= O(br) * O((\frac{m}{b})^2) + O(b^{3/2} r^{1/2}) * O(\frac{m}{b}) \\
&= O(m^2 r/b + m(br)^{1/2}) \\
&= \mathbf{O}(\mathbf{m}^{\mathbf{4/3}} \mathbf{r}^{\mathbf{2/3}}) \text{ for } b = (m^2 r)^{1/3}
\end{aligned}
$$

Theo Mary (contact: theo.mary@manchester.ac.uk)

Complexity of the two-level BLR format

Assume all off-diagonal blocks are low-rank. Then:



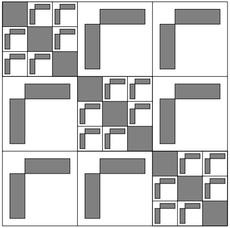
$$
\begin{aligned}
Storage &= cost_{LR} * nb_{LR} + cost_{BLR} * nb_{BLR} \\
&= O(br) * O((\frac{m}{b})^2) + O(b^{3/2} r^{1/2}) * O(\frac{m}{b}) \\
&= O(m^2 r/b + m(br)^{1/2}) \\
&= \mathbf{O(m^{4/3} r^{2/3})} \text{ for } b = (m^2 r)^{1/3}
\end{aligned}
$$

Similarly, we can prove:

$$FlopLU = \mathbf{O(m^{5/3} r^{4/3})} \text{ for } b = (m^2 r)^{1/3}$$

Result holds if a **constant** number of off-diag. blocks is BLR.

Assume all off-diagonal blocks are low-rank. Then:



$$
\begin{aligned}
Storage &= cost_{LR} * nb_{LR} + cost_{BLR} * nb_{BLR} \\
&= O(br) * O((\frac{m}{b})^2) + O(b^{3/2} r^{1/2}) * O(\frac{m}{b}) \\
&= O(m^2 r/b + m(br)^{1/2}) \\
&= \mathbf{O(m^{4/3} r^{2/3})} \text{ for } b = (m^2 r)^{1/3}
\end{aligned}
$$

Similarly, we can prove:

$$FlopLU = \mathbf{O(m^{5/3} r^{4/3})} \text{ for } b = (m^2 r)^{1/3}$$

Result holds if a **constant** number of off-diag. blocks is BLR.

		FR	BLR	2-BLR	...	$\mathcal{H}$
storage	dense	$O(m^2)$	$O(m^{1.5})$	$O(m^{1.33})$	...	$O(m \log m)$
	sparse	$O(n^{1.33})$	$O(n \log n)$	$O(n)$	...	$O(n)$
flop LU	dense	$O(m^3)$	$O(m^2)$	$O(m^{1.66})$	...	$O(m \log^3 m)$
	sparse	$O(n^2)$	$O(n^{1.33})$	$O(n^{1.11})$	...	$O(n)$

Multilevel BLR complexity

Main result

For $b = m^{\ell/(\ell+1)} r^{1/(\ell+1)}$, the $\ell-$level complexities are:

$$\mathit{Storage} = \mathbf{O}(\mathbf{m}^{(\ell+2)/(\ell+1)} \mathbf{r}^{\ell/(\ell+1)})$$

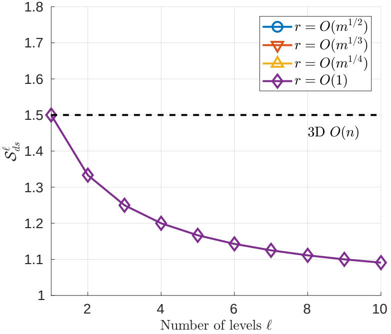
$$\mathit{FlopLU} = \mathbf{O}(\mathbf{m}^{(\ell+3)/(\ell+1)} \mathbf{r}^{2\ell/(\ell+1)})$$

Proof: by induction. $\square$

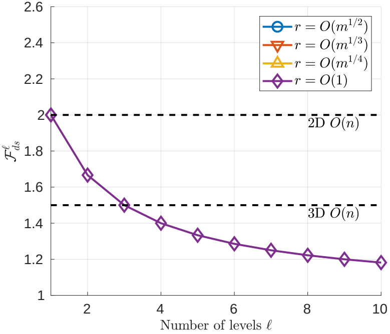
- Simple way to finely control the desired complexity
- Block size $b \propto O(m^{\ell/(\ell+1)}) \ll O(m)$
  $\Rightarrow$ may be efficiently processed in shared-memory
- Number of blocks per row/column $\propto O(m^{1/(\ell+1)}) \gg O(1)$
  $\Rightarrow$ flexibility to distribute data in parallel

Theo Mary (contact: theo.mary@manchester.ac.uk)

Storage



Flop LU

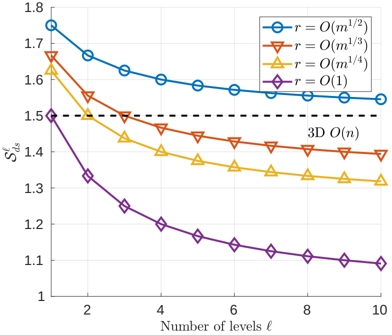


- If $r = O(1)$, can achieve $O(n)$ storage complexity with only two levels and $O(n \log n)$ flop complexity with three levels

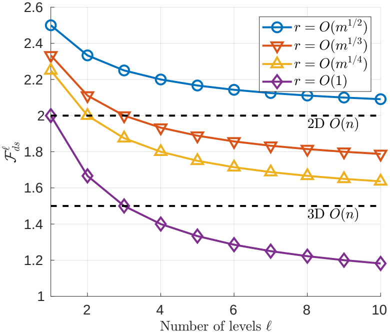
Theo Mary (contact: theo.mary@manchester.ac.uk)

Influence of the number of levels ℓ

Storage



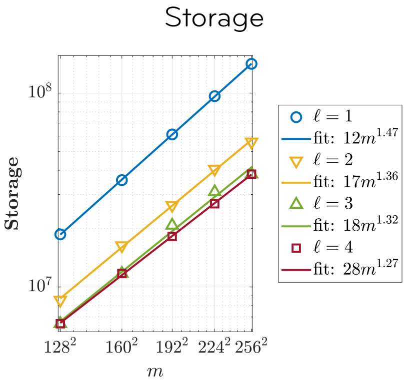
Flop LU

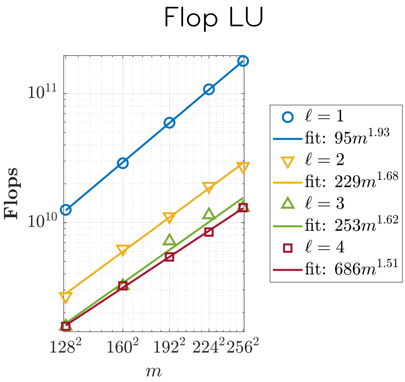


- If $r = O(1)$, can achieve $O(n)$ storage complexity with only two levels and $O(n \log n)$ flop complexity with three levels
- For higher ranks, optimal sparse complexity is not attainable with constant ℓ **but** improvement rate is rapidly decreasing: **the first few levels achieve most of the asymptotic gain**

Theo Mary (contact: theo.mary@manchester.ac.uk)

Numerical experiments (Poisson)





- Experimental complexity in relatively good agreement with theoretical one
- Asymptotic gain decreases with levels

Conclusion

Conclusions and perspectives

A new multilevel format to...

- Finely control desired complexity between BLR's and $\mathcal{H}$'s
- Strike a balance between BLR's simplicity and $\mathcal{H}$'s complexity
- Trade off $\mathcal{H}$'s nearly linear dense complexity and still achieve $\mathcal{C}_{sparse} = O(n)$

Future work: high-performance implementation

- Implementation of the MBLR format in a parallel, algebraic, general purpose sparse solver (e.g. MUMPS)

Theo Mary (contact: theo.mary@manchester.ac.uk)

Thank you for your attention

Slides and paper available here:

personalpages.manchester.ac.uk/staff/theo.mary/