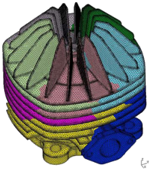
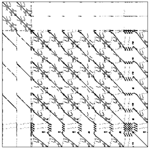
Bridging the gap between flat and hierarchical low-rank matrix formats

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

SIAM CSE 2019, Spokane, Feb. 25th - March 1st

 $\rightarrow$  $\rightarrow$ 

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!

Need to **reduce the asymptotic complexity**

- converting complexity gains into **real performance gains**
- and reach **application required accurary**

Block Low-Rank general context and main features

- Applicative context: discretized PDEs, integral equations
- Compute an approximate factorization $\mathbf{A} \approx \mathbf{L}_\varepsilon \mathbf{U}_\varepsilon$ at accuracy ε controlled by the user

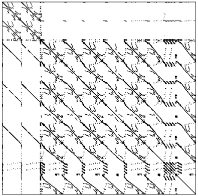
Block Low-Rank[1] (BLR)

- Flat and simple format
  - Algebraic robust solver;
  - Compatible with the numerical features of a general solver (such as partial threshold pivoting for stability)
- *Work supported by PhD theses from University of Toulouse, C. Weisbecker (2010-2013, supported by EDF) and T. Mary (2014-2017)*

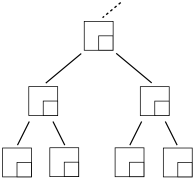
$\Rightarrow$ Many representations: Recursive $\mathcal{H}, \mathcal{H}^2$ [Bebendof, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...], BLR ...

[1] *[Amestoy, Ashcraft, Boiteau, Buttari, L'Excellent, and Weisbecker, SIAM J. Sci. Comput., 2015]*

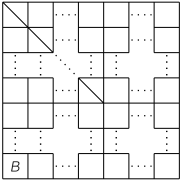
Block Low-Rank Multifrontal feature: principle



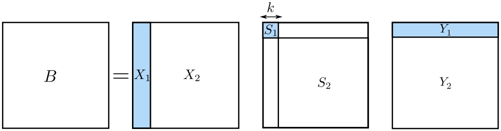
$\Rightarrow$



Elimination tree

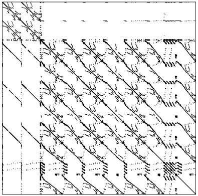


$\Rightarrow$

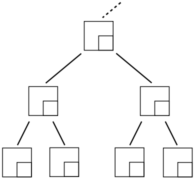


Singular value decomposition (SVD) of each block $B \Rightarrow B = X_1 S_1 Y_1 + X_2 S_2 Y_2$

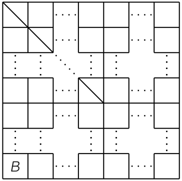
Block Low-Rank Multifrontal feature: principle



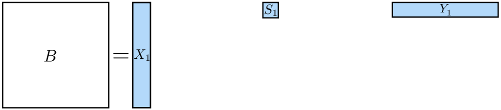
$\Rightarrow$



Elimination tree



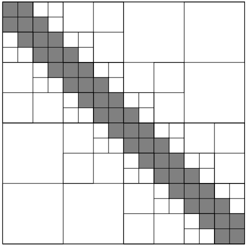
$\Rightarrow$



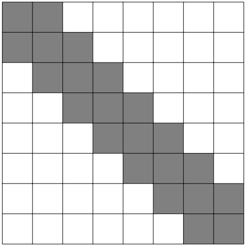
rank $k(\varepsilon)$: $B = X_1 S_1 Y_1 \cancel{+ X_2 S_2 Y_2}$

$\|E\|_2 = \|X_2 S_2 Y_2\|_2 = \sigma_{k+1} \leq \varepsilon$

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$ matrix



BLR matrix

- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure

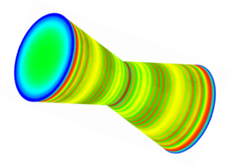
- Theoretical complexity can be as low as $O(n^{4/3})$
- Simpler structure

BLR makes easier to preserve the numerical features of a direct solver and compromises well complexity, accuracy and performance

BLR complexity[2] (Poisson $n = N^3$, n: matrix size, N: grid size)

- Operations for sparse factorization $\mathcal{O}\left(n^2\right) \rightarrow \mathcal{O}\left(n^{4/3}\right)$
- Convert it into performance gains, not straightforward[3]

Required accuracy: 10^{-9}



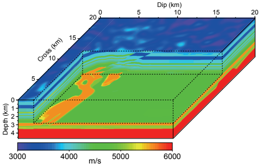
Structural mechanics

$n = 8M$

Flop Ratio=17

Time Ratio= 6

Required accuracy: 10^{-3}



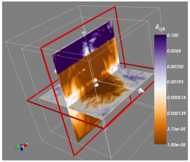
Seismic imaging

$n = 17M$

Flop Ratio=27

Time Ratio= 7

Required accuracy: 10^{-7}



Electromagnetism

$n = 21M$

Flop Ratio=65

Time Ratio=19

---

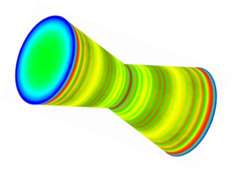
[2] proved in *[Amestoy, Buttari, L'Excellent, Mary, SIAM J. Sci. Comput. 2017]*

[3] *[Amestoy, Buttari, L'Excellent, Mary, Trans. on Math. Soft. 2018], 24 Haswell cores*

BLR complexity[2] (Poisson $n = N^3$, n: matrix size, N: grid size)

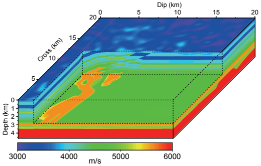
- Operations for sparse factorization $\mathcal{O}\left(n^2\right) \to \mathcal{O}\left(n^{4/3}\right)$
- Convert it into performance gains, not straightforward[3]

Required accuracy: 10^{-9}



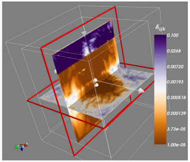
Structural mechanics
$n = 8M$
Flop Ratio=17
Time Ratio= 6

Required accuracy: 10^{-3}



Seismic imaging
$n = 17M$
Flop Ratio=27
Time Ratio= 7

Required accuracy: 10^{-7}



Electromagnetism
$n = 21M$
Flop Ratio=65
Time Ratio=19

Can we reduce complexity and preserve performance ?

[2] proved in *[Amestoy, Buttari, L'Excellent, Mary, SIAM J. Sci. Comput. 2017]*

[3] *[Amestoy, Buttari, L'Excellent, Mary, Trans. on Math. Soft. 2018], 24 Haswell cores*

Outline

1. *Why is sparse factorization a better playground for BLR than dense factorization ?*
2. *How to do the minimum to reach a target asymphthotic complexity?*

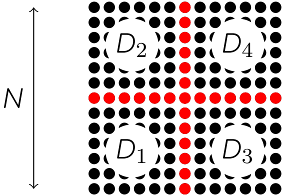
   Multilevel BLR (**MBLR**):
   - Complexity analysis
   - Numerical results
3. Concluding remarks

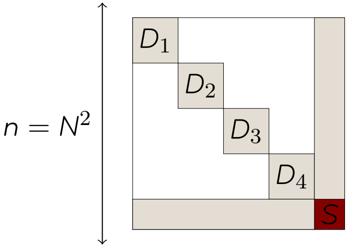
Preprint

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).

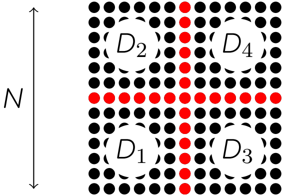
Sparse factorization a better playground for BLR than dense factorization?

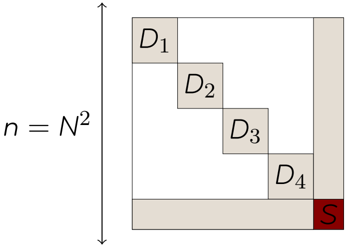
From dense to sparse: nested dissection

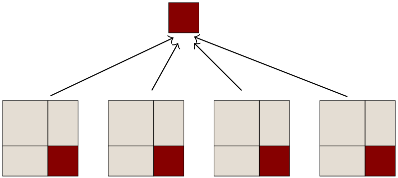




From dense to sparse: nested dissection







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

Nested dissection complexity formulas

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

Nested dissection complexity formulas

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

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

Nested dissection complexity formulas

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

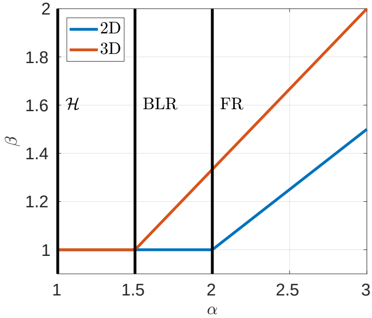
3D: $\mathcal{C}_{sparse} = \sum_{\ell=0}^{\log N} 8^{\ell} \mathcal{C}_{dense}(\frac{N^2}{4^{\ell}}) \quad \to$ common ratio $2^{3-2\alpha}$

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

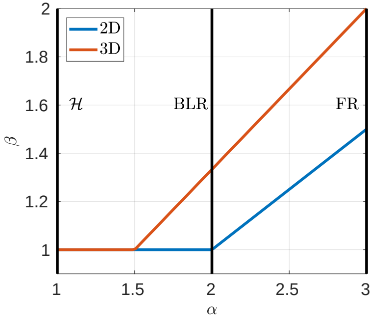
2D		3D	
	$\mathcal{C}_{sparse}(n)$		$\mathcal{C}_{sparse}(n)$
$\alpha > 2$	$O(n^{\alpha/2})$	$\alpha > 1.5$	$O(n^{2\alpha/3})$
$\alpha = 2$	$O(n \log n)$	$\alpha = 1.5$	$O(n \log n)$
$\alpha < 2$	$O(n)$	$\alpha < 1.5$	$O(n)$

$$\mathcal{C}_{dense} = O(m^{\alpha}) \Rightarrow \mathcal{C}_{sparse} = O(n^{\beta})$$

Storage

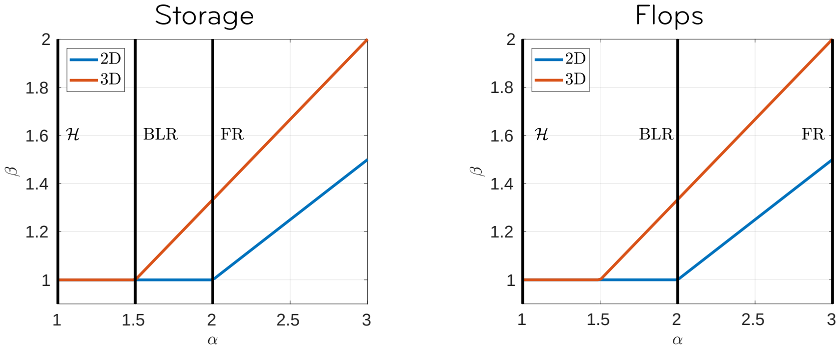


Flops



Bridging the gap between flat and hierarchical formats

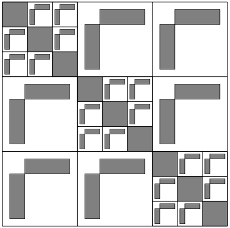
$$\mathcal{C}_{dense} = O(m^{\alpha}) \Rightarrow \mathcal{C}_{sparse} = O(n^{\beta})$$



Key motivation: $\mathcal{C}_{dense} < O(m^2)$ (2D) or $O(m^{1.5})$ (3D) is enough to get $O(n)$ sparse complexity!

The multilevel BLR (MBLR) format

Complexity of the two-level BLR format



Two-level BLR format: replace full-rank blocks by BLR matrices

For $b = (m^2 r)^{1/3}$:

$$Storage = \mathbf{O(m^{4/3} r^{2/3})}$$

$$FlopLU = \mathbf{O(m^{5/3} r^{4/3})}$$

		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:

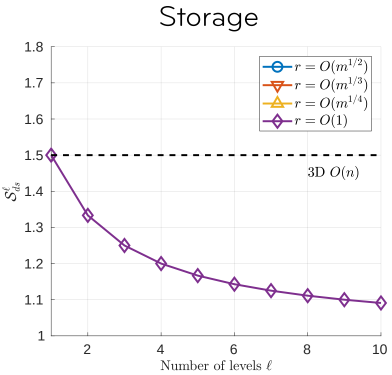
$$Storage = \mathbf{O}(\mathbf{m}^{(\ell+2)/(\ell+1)} \mathbf{r}^{\ell/(\ell+1)})$$

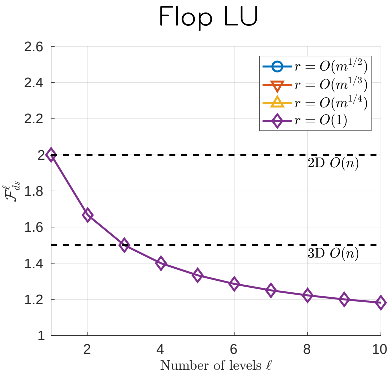
$$FlopLU = \mathbf{O}(\mathbf{m}^{(\ell+3)/(\ell+1)} \mathbf{r}^{2\ell/(\ell+1)})$$

Proof: by induction. $\square$

- Simple way to finely control the desired complexity
- Block size $b \propto O(m^{1-1/(\ell+1)}) \ll O(m)$
  $\Rightarrow$ larger blocks that can be efficiently processed in shared-memory

Influence of the number of levels ℓ

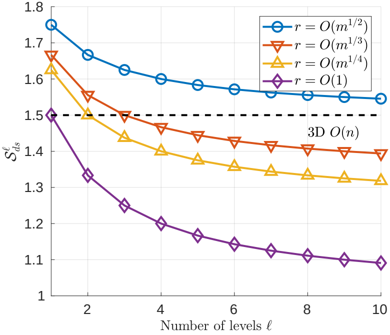




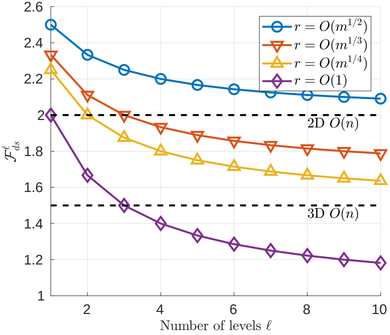
- If $r = O(1)$, can achieve $O(n)$ storage complexity with only two levels and $O(n \log n)$ flop complexity with three levels

Influence of the number of levels ℓ

Storage

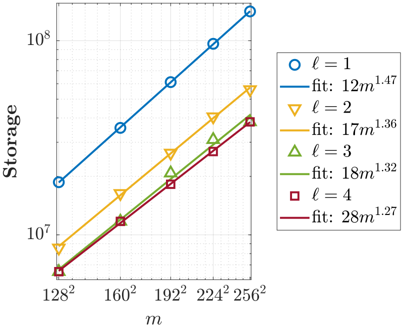


Flop LU

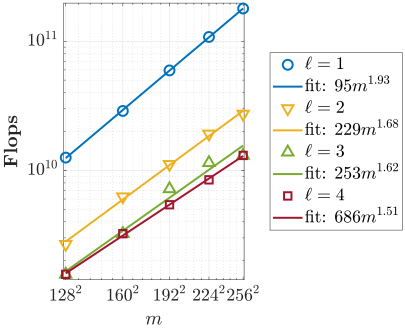


- If $r = O(1)$, can achieve $O(n)$ storage complexity with only two levels and $O(n \log n)$ flop complexity with three levels
- For higher ranks, improvement rate rapidly decreases: **the first few levels achieve most of the asymptotic gain**

Storage



Flop LU



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

Concluding remarks

Conclusions and perspectives

A new multilevel format to...

- Finely control desired complexity between BLR's and $\mathcal{H}$'s
- Find 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

- Implementation of the MBLR format in a parallel, algebraic, general purpose sparse solver (e.g. MUMPS)
- Algorithmic work to reach high performance on parallel architectures (just as it was needed for BLR)