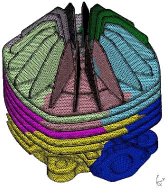
Complexity, Scalability, and Stability of Block Low-Rank Solvers

Theo Mary
University of Manchester, School of Mathematics

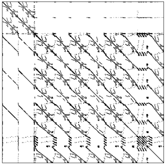
LIP, ENS Lyon, 11 December 2018

M\cr NA
Manchester Numerical Analysis

Context



→



→

Linear system $Ax = b$

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

Large, sparse matrices

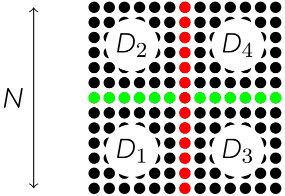
Matrix A is sparse (many zeros) but also large (10^6–10^9 unknowns)

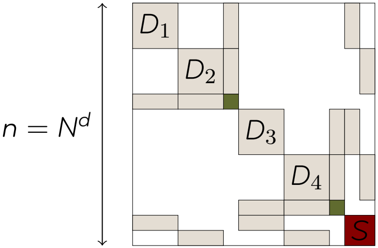
Direct methods

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

☺ Numerically reliable ☹ Computational cost

Structural sparsity



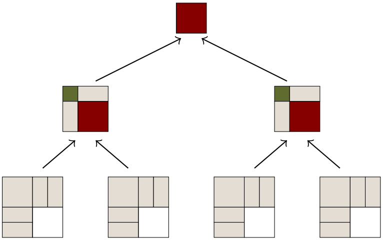


2D problem complexity

- Flops: $O(n^3) \to O(n^{3/2})$
- Storage: $O(n^2) \to O(n \log n)$

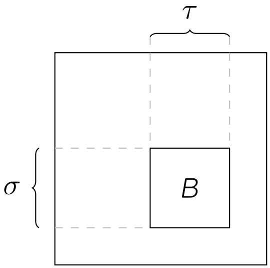
3D problem complexity

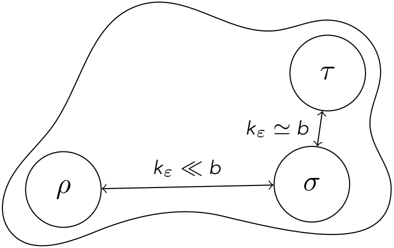
- Flops: $O(n^3) \to O(n^2)$
- Storage: $O(n^2) \to O(n^{4/3})$



Data sparsity

In many cases of interest the matrix has a block low-rank structure





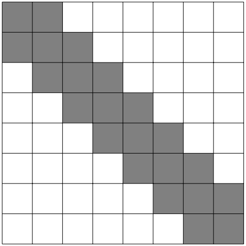
A block B represents the interaction between two subdomains.

Far away subdomains $\Rightarrow$ block of low numerical rank:

$$\underset{b \times b}{B} \approx \underset{b \times k_\varepsilon}{X} \underset{k_\varepsilon \times b}{Y^T}$$

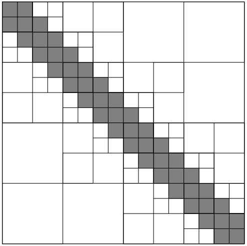
with $k_\varepsilon \ll b$ such that $\|B - XY^T\| \leq \varepsilon$

How to choose a good block partitioning of the matrix?



BLR matrix

- Superlinear complexity
- Simple, flat structure



$\mathcal{H}$-matrix

- Nearly linear complexity
- Complex, hierarchical structure

Outline

1. **Asymptotic complexity of BLR factorization**

> 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).

> 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).

2. **Performance and scalability of BLR solvers**

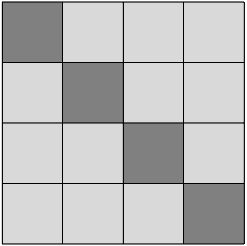
> P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*. ACM Trans. Math. Soft. (2018).

3. **Rounding error analysis of BLR factorization**
4. **Low-accuracy BLR preconditioners**

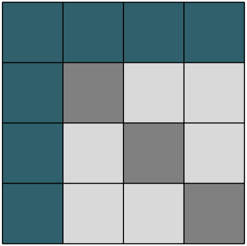
> N. Higham and T. Mary. *A New Preconditioner that Exploits Low-Rank Approximations to Factorization Error*. SIAM J. Sci. Comp (2018).

Asymptotic complexity of BLR factorization

BLR factorization: standard FCU variant

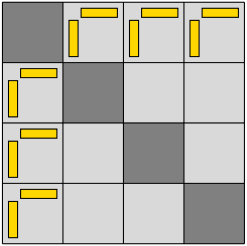


- FCU



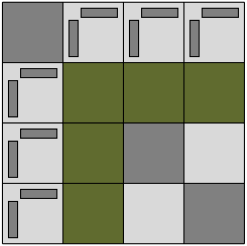
- FCU (Factor,
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



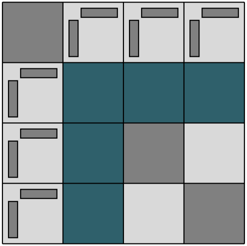
- FCU (Factor, Compress,
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



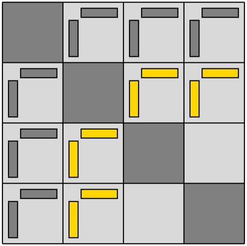
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



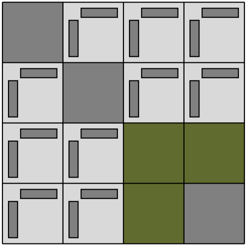
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



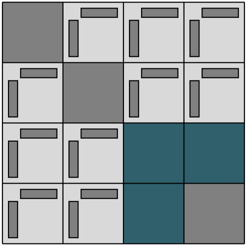
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



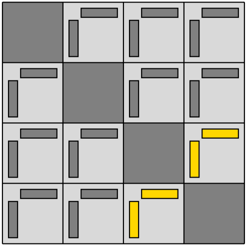
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



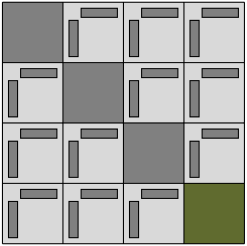
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



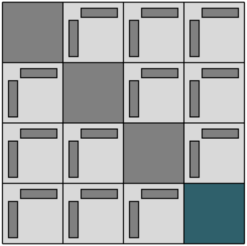
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



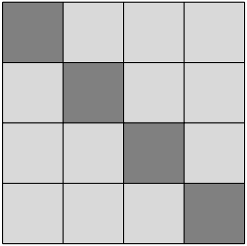
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

BLR factorization: standard FCU variant



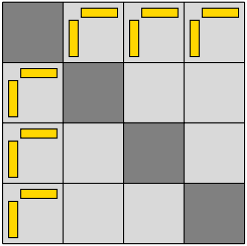
- FCU (Factor, Compress, Update)
- Easy to handle numerical pivoting

CFU factorization variant



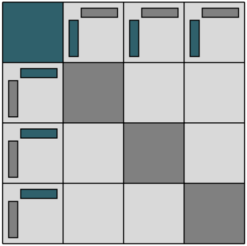
- CFU

CFU factorization variant



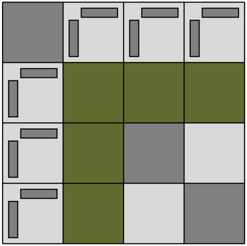
- CFU (Compress,

CFU factorization variant



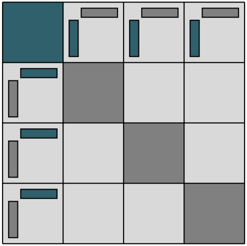
- CFU (Compress, Factor,
- Factor step is performed on compressed blocks ⇒ reduced flops

CFU factorization variant



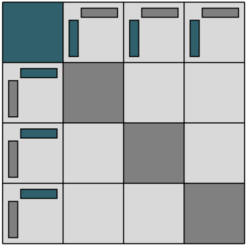
- CFU (Compress, Factor, Update)
- Factor step is performed on compressed blocks ⇒ reduced flops

CFU factorization variant



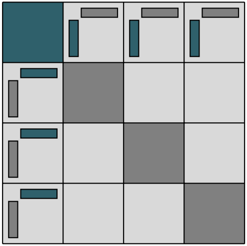
- CFU (Compress, Factor, Update)
- Factor step is performed on compressed blocks ⇒ reduced flops
- How can we handle numerical pivoting?

CFU factorization variant



- CFU (Compress, Factor, Update)
- Factor step is performed on compressed blocks ⇒ reduced flops
- How can we handle numerical pivoting?
  - Restricting pivot choice to diagonal block is acceptable (in combination with a pivot delaying strategy)

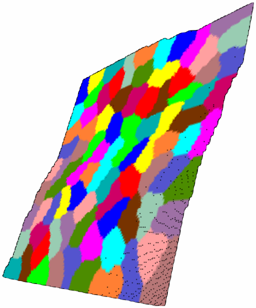
CFU factorization variant



- CFU (Compress, Factor, Update)
- Factor step is performed on compressed blocks ⇒ reduced flops
- How can we handle numerical pivoting?
  - Restricting pivot choice to diagonal block is acceptable (in combination with a pivot delaying strategy)
  - Must still check entries in off-diagonal blocks: can be estimated from entries in low-rank blocks

Complexity of the dense BLR factorization

Complexity analysis based on $\mathcal{H}$ matrix theory and requires the important assumption that the number of full-rank blocks per row/column is constant $\Rightarrow$ can be guaranteed with an adequate clustering (so-called BLR admissibility condition)



Then, for a $m \times m$ dense matrix with blocks of rank r:

- Storage: $O(m^2) \to O(m^{3/2}r^{1/2})$
- Flops LU: $O(m^3) \to O(m^{7/3}r^{2/3})$ (FCU) $\to O(m^2 r)$ (CFU)

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).

Complexity of the sparse BLR factorization

		storage	flops
dense	FR	$O(m^2)$	$O(m^3)$
	BLR	$O(m^{3/2})$	$O(m^2)$
	$\mathcal{H}$	$O(m \log m)$	$O(m \log^2 m)$

(assuming $r = O(1)$)

Complexity of the sparse BLR factorization

		storage	flops
dense	FR	$O(m^2)$	$O(m^3)$
	BLR	$O(m^{3/2})$	$O(m^2)$
	$\mathcal{H}$	$O(m \log m)$	$O(m \log^2 m)$
sparse 2D	FR	$O(n \log n)$	$O(n^{3/2})$
	BLR	$O(n)$	$O(n \log n)$
	$\mathcal{H}$	$O(n)$	$O(n)$

(assuming $r = O(1)$)

- In a **2D** world hierarchical matrices would not be needed

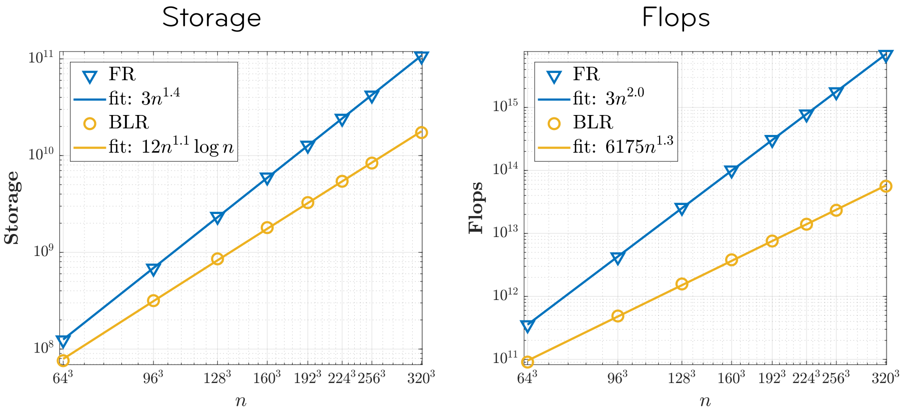
Complexity of the sparse BLR factorization

		storage	flops
dense	FR	$O(m^2)$	$O(m^3)$
	BLR	$O(m^{3/2})$	$O(m^2)$
	$\mathcal{H}$	$O(m \log m)$	$O(m \log^2 m)$
sparse 2D	FR	$O(n \log n)$	$O(n^{3/2})$
	BLR	$O(n)$	$O(n \log n)$
	$\mathcal{H}$	$O(n)$	$O(n)$
sparse 3D	FR	$O(n^{4/3})$	$O(n^2)$
	BLR	$O(n \log n)$	$O(n^{4/3})$
	$\mathcal{H}$	$O(n)$	$O(n)$

(assuming $r = O(1)$)

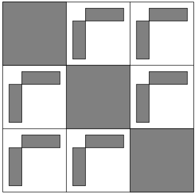
- In a **2D** world hierarchical matrices would not be needed
- Superlinear complexities in **3D**

Experimental complexity fit: 3D Poisson ($\varepsilon = 10^{-10}$)

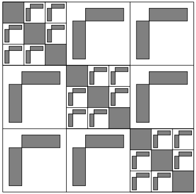


- Good agreement with theoretical complexity:
  - Storage: $O(n\log n) \to O(n^{1.1}\log n)$
  - Flops: $O(n^{4/3}) \to O(n^{1.3})$

Two-level BLR format



Two-level BLR format

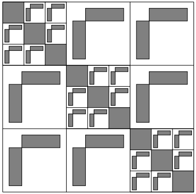


Key idea: replace full-rank blocks by their BLR approximation ⇒ two-level BLR matrix

Technically hierarchical, but much simpler

Key idea: replace full-rank blocks by their BLR approximation $\Rightarrow$ two-level BLR matrix

Technically hierarchical, but much simpler



Storage complexity (assuming $r = O(1)$):

	FR	BLR	2-BLR	...	$\mathcal{H}$
dense	$O(m^2)$	$O(m^{1.5})$	$O(m^{1.33})$	...	$O(m \log m)$
sparse (2D)	$O(n \log n)$	$O(n)$	$O(n)$	...	$O(n)$
sparse (3D)	$O(n^{1.33})$	$O(n \log n)$	$O(n)$	...	$O(n)$

Flop complexity (assuming $r = O(1)$):

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

Multilevel BLR format

Multilevel BLR (MBLR) format: recursively refine full-rank blocks up to a constant number of levels ℓ

MBLR complexity

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

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

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).

Multilevel BLR format

Multilevel BLR (MBLR) format: recursively refine full-rank blocks up to a constant number of levels ℓ

MBLR complexity

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

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

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).

Flop complexity (assuming $r = O(1)$):

	$\ell = 1$	$\ell = 2$	Hierar.
Dense	$O(m^2)$	$O(m^{1.66})$	$O(m \log^2 m)$
Sparse (3D)	$O(n^{1.33})$	$O(n^{1.11})$	$O(n)$

Multilevel BLR format

Multilevel BLR (MBLR) format: recursively refine full-rank blocks up to a constant number of levels ℓ

MBLR complexity

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

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

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).

Flop complexity (assuming $r = O(1)$):

	$\ell = 1$	$\ell = 2$	$\ell = 3$	Hierar.
Dense	$O(m^2)$	$O(m^{1.66})$	$O(m^{1.5})$	$O(m \log^2 m)$
Sparse (3D)	$O(n^{1.33})$	$O(n^{1.11})$	$O(n \log n)$	$O(n)$

Multilevel BLR (MBLR) format: recursively refine full-rank blocks up to a constant number of levels ℓ

MBLR complexity

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

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

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).

Flop complexity (assuming $r = O(1)$):

	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	Hierar.
Dense	$O(m^2)$	$O(m^{1.66})$	$O(m^{1.5})$	$O(m^{1.4})$	$O(m \log^2 m)$
Sparse (3D)	$O(n^{1.33})$	$O(n^{1.11})$	$O(n \log n)$	$O(n)$	$O(n)$

Multilevel BLR format

Multilevel BLR (MBLR) format: recursively refine full-rank blocks up to a constant number of levels ℓ

MBLR complexity

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

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

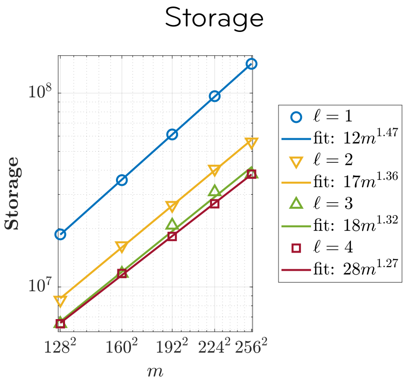
P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. *Bridging the gap between flat and hierarchical low-rank matrix formats: the multilevel BLR format*. Submitted (2018).

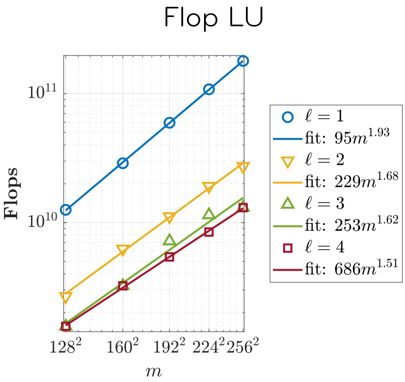
Flop complexity (assuming $r = O(1)$):

	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	Hierar.
Dense	$O(m^2)$	$O(m^{1.66})$	$O(m^{1.5})$	$O(m^{1.4})$	$O(m \log^2 m)$
Sparse (3D)	$O(n^{1.33})$	$O(n^{1.11})$	$O(n \log n)$	$O(n)$	$O(n)$

With $r = O(1)$ only 4 levels are enough. With larger ranks more levels needed but **gain from adding more levels decreases rapidly**

Numerical experiments (3D Poisson)



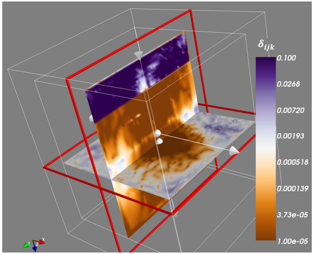


- Experimental complexity in relatively good agreement with theoretical one
- Asymptotic gain decreases with levels

Performance and scalability of BLR solvers

Shared-memory performance analysis: an example

Matrix S3
Double complex (z) symmetric
Electromagnetics application (CSEM)
3.3 millions unknowns
Required accuracy: $\varepsilon = 10^{-7}$



D. Shantsev, P. Jaysaval, S. Kethulle de Ryhove, P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. *Large-scale 3D EM modeling with a Block Low-Rank multifrontal direct solver*. Geophys. J. Int (2017).

	flops ($\times 10^{12}$)	time (1 core)	time (24 cores)
FR	78.0	7390	509
BLR	10.2	2242	307
ratio	7.7	3.3	1.7

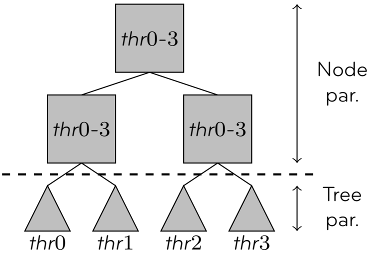
7.7 gain in flops only translated to a **1.7** gain in time:
Can we do better?

Variant name	time	FR/BLR ratio
Full-Rank	509	
BLR (FCU)	307	1.7

Improving the performance of BLR factorization

Tree parallelism improves performance by reducing the relative cost of the fronts at the bottom of the tree, which achieve poor compression

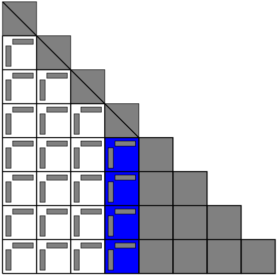
Variant name	time	FR/BLR ratio
Full-Rank	509	
+Tree par.	418	
BLR (FCU)	307	1.7
+Tree par.	221	1.9



Improving the performance of BLR factorization

Left-looking FCU improves performance thanks to a left-looking approach which reduces memory transfers

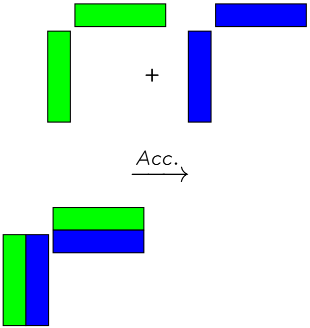
Variant name	time	FR/BLR ratio
Full-Rank	509	
+Tree par.	418	
BLR (FCU)	307	1.7
+Tree par.	221	1.9
+Left-looking	175	2.4



Improving the performance of BLR factorization

LUA improves performance because it accumulates multiple low-rank updates and applies them at once increasing the granularity of operations

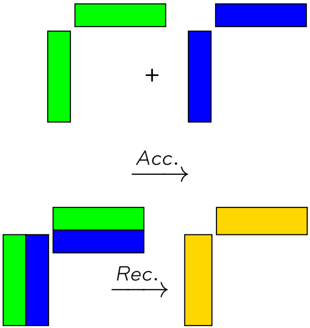
Variant name	time	FR/BLR ratio
Full-Rank	509	
+Tree par.	418	
BLR (FCU)	307	1.7
+Tree par.	221	1.9
+Left-looking	175	2.4
+Accumulation	167	2.5



Improving the performance of BLR factorization

LUAR reduces complexity because recompresses accumulated updates on the fly

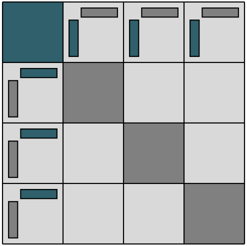
Variant name	time	FR/BLR ratio
Full-Rank	509	
+Tree par.	418	
BLR (FCU)	307	1.7
+Tree par.	221	1.9
+Left-looking	175	2.4
+Accumulation	167	2.5
+Recompression	160	2.6



Improving the performance of BLR factorization

CFU reduces complexity because solve operations are also done in low-rank

Variant name	time	FR/BLR ratio
Full-Rank	509	
+Tree par.	418	
BLR (FCU)	307	1.7
+Tree par.	221	1.9
+Left-looking	175	2.4
+Accumulation	167	2.5
+Recompression	160	2.6
+CFU	111	3.8



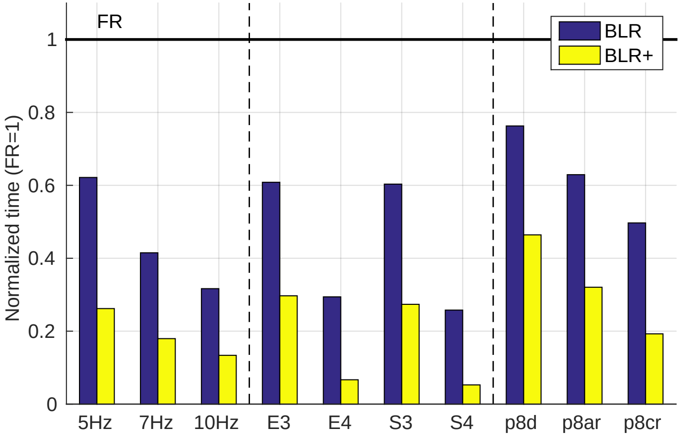
Improving the performance of BLR factorization

Variant name	time	FR/BLR ratio
Full-Rank	509	
+Tree par.	418	
BLR (FCU)	307	1.7
+Tree par.	221	1.9
+Left-looking	175	2.4
+Accumulation	167	2.5
+Recompression	160	2.6
+CFU	111	3.8

Converting the theoretical flop reduction into **actual time gains on modern architectures** requires careful algorithmic work

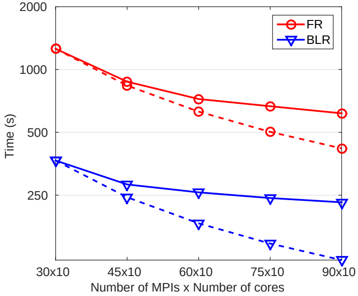
Results with the BLR MUMPS solver:

P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*. ACM Trans. Math. Soft. (2018).



Distributed-memory performance results

Results on $300 \rightarrow 900$ cores
(eos supercomputer, CALMIP)



Matrix 10Hz
Single complex (c) unsymmetric
Seismic imaging application (FWI)
17 millions unknowns
Required accuracy: $\varepsilon = 10^{-3}$

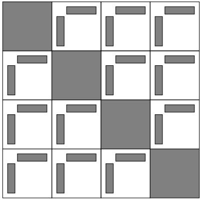
P. Amestoy, R. Brossier, A. Buttari, J.-Y. L'Excellent, T. Mary, L. Métivier, A. Miniussi, and S. Operto. *Fast 3D frequency-domain full waveform inversion with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea*. Geophysics (2016).

How to improve the scalability of the BLR factorization?
Two main difficulties:

- **Higher weight of communications:** flops reduced by **13** but volume of communications only by **2**
- **Unpredictability of compression:** more difficult to design good mapping and scheduling strategies

Rounding error analysis of BLR factorization

Why we need an error analysis



Each off-diagonal block B is approximated by a low-rank matrix $\widetilde{B}$ such that $\|B - \widetilde{B}\| \le \varepsilon \|B\|$
$\Rightarrow \|A - A_\varepsilon\| \le \varepsilon \|A\|$ with good norm choice
However:
$\|A - L_\varepsilon U_\varepsilon\| \ne \varepsilon$ because of rounding errors
$\Rightarrow$ **What is the overall accuracy $\|A - L_\varepsilon U_\varepsilon\|$?**

- Can we prove that $\|A - L_\varepsilon U_\varepsilon\| = O(\varepsilon)$? What is the role of the unit roundoff u?
- What is the error growth, i.e., how does the error depend on the matrix size n?
- How do the different variants (FCU, CFU, etc.) compare?
- Should we use an absolute threshold ($\|B - \widetilde{B}\| \le \varepsilon$) or a relative one ($\|B - \widetilde{B}\| \le \varepsilon \|B\|$)?

Main result: statement

Reminder

The full-rank LU factorization of $A \in \mathbb{R}^{n \times n}$ satisfies

$$\|A - LU\| \leq nu\|L\|\|U\| + O(u^2)$$

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n \times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \leq (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

The proof is quite technical and based on *Stability of Block Algorithms with Fast Level-3 BLAS* (Demmel and Higham, 1992)

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \leq (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

Main result: comments

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n \times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

- $\|L\|\|U\| \le n^2 \rho_n \|A\|$ where ρ_n is the growth factor
  $\Rightarrow$ with partial pivoting, the BLR factorization is stable!

Main result: comments

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n \times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

- $\|L\|\|U\| \le n^2 \rho_n \|A\|$ where ρ_n is the growth factor
  $\Rightarrow$ with partial pivoting, the BLR factorization is stable!
- Usually $\varepsilon \gg u$:

$\Rightarrow$ Role of u is limited

$\Rightarrow$ Very slow error growth

$\Rightarrow$ Usage of fast matrix arithmetic
may be stable in BLR

Main result: comments

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

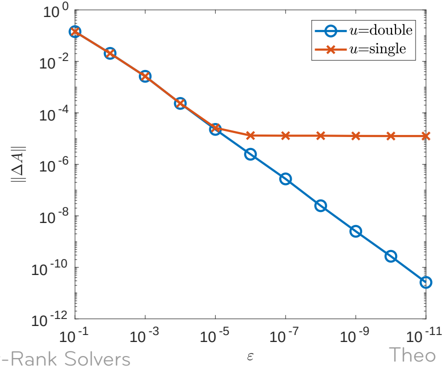
$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

- $\|L\|\|U\| \le n^2\rho_n\|A\|$ where ρ_n is the growth factor
  $\Rightarrow$ with partial pivoting, the BLR factorization is stable!
- Usually $\varepsilon \gg u$:

$\Rightarrow$ Role of u is limited

$\Rightarrow$ Very slow error growth

$\Rightarrow$ Usage of fast matrix arithmetic may be stable in BLR



Main result: comments

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

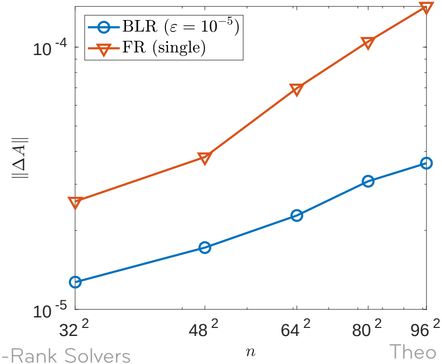
$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

- $\|L\|\|U\| \le n^2 \rho_n \|A\|$ where ρ_n is the growth factor
  $\Rightarrow$ with partial pivoting, the BLR factorization is stable!
- Usually $\varepsilon \gg u$:

$\Rightarrow$ Role of u is limited

$\Rightarrow$ Very slow error growth

$\Rightarrow$ Usage of fast matrix arithmetic may be stable in BLR



Main result: comments

Main result

The FCU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

- $\|L\|\|U\| \le n^2 \rho_n \|A\|$ where ρ_n is the growth factor
  $\Rightarrow$ with partial pivoting, the BLR factorization is stable!
- Usually $\varepsilon \gg u$:

$\Rightarrow$ Role of u is limited

For example with Strassen's algorithm, $nu \to n^{\log_2 12} u \approx n^{3.6} u$

$\Rightarrow$ Very slow error growth

$\Rightarrow$ Usage of fast matrix arithmetic may be stable in BLR

Ongoing work with C.-P. Jeannerod, C. Pernet, and D. Roche: *Exploiting fast matrix arithmetic within BLR factorizations*:
$O(n^2)$ complexity $\to O(n^{(\omega+1)/2})$
($\approx O(n^{1.9})$ for Strassen)

Relative vs absolute threshold

Theorem

The FCU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with absolute threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \theta\varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

where $\theta = \sqrt{n/b - 1}\sum_{i=1}^{n/b} \|L_{ii}\| + \|U_{ii}\|$

The BLR factorization with absolute threshold

- ☹ Has a faster error growth
- ☹ Is scaling-dependent

Relative vs absolute threshold

Theorem

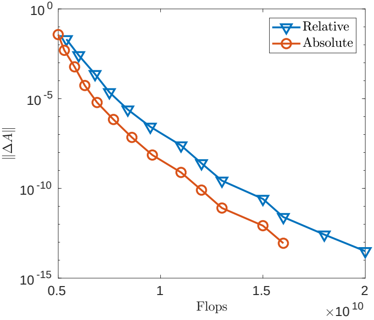
The FCU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with absolute threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \theta\varepsilon)\|L\|\|U\| + O(u\varepsilon) + O(u^2)$$

where $\theta = \sqrt{n/b - 1}\sum_{i=1}^{n/b} \|L_{ii}\| + \|U_{ii}\|$

The BLR factorization with absolute threshold

- ☹ Has a faster error growth
- ☹ Is scaling-dependent
- ☺ Is more efficient in practice



Theorem

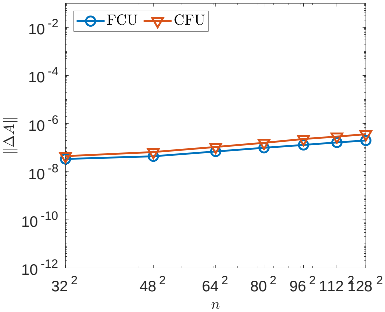
The CFU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(\kappa(A)u\varepsilon) + O(u^2)$$

Theorem

The CFU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

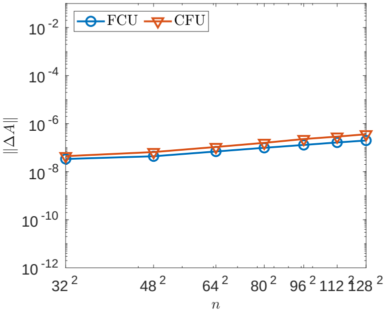
$$\|A - L_\varepsilon U_\varepsilon\| \leq (nu + \varepsilon)\|L\|\|U\| + O(\kappa(A)u\varepsilon) + O(u^2)$$

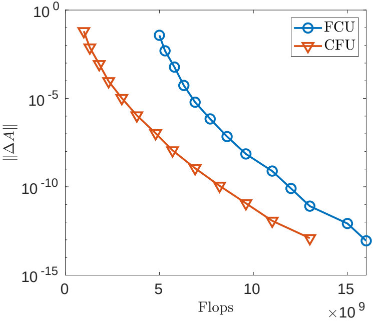


Theorem

The CFU BLR factorization of $A \in \mathbb{R}^{n\times n}$ with relative threshold ε satisfies

$$\|A - L_\varepsilon U_\varepsilon\| \le (nu + \varepsilon)\|L\|\|U\| + O(\kappa(A)u\varepsilon) + O(u^2)$$

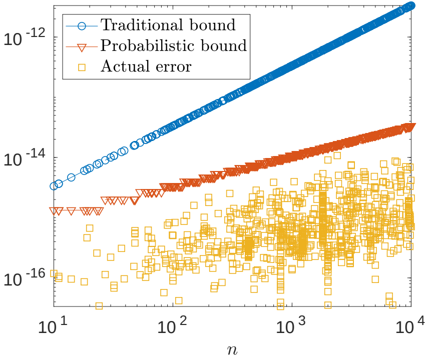




Traditional bound: $|A - LU| \le nu|L||U|$

Traditional bound offers no accuracy guarantees when nu is large
$\Rightarrow$ important issue for large-scale, low-precision computations

Probabilistic bound: $|A - LU| \le \lambda\sqrt{n}u|L||U|$ with high probability



Solution of $Ax = b$, for 943 matrices from the SuiteSparse collection

N. Higham and T. Mary. *A New Approach to Probabilistic Rounding Error Analysis*. Submitted (2018).

Probabilistic rounding error analysis could be of special relevance for low-rank solvers: $nu + \varepsilon \to \lambda\sqrt{n}u + \varepsilon$?

Low-accuracy BLR preconditioners

Low-accuracy BLR preconditioners: storage

BLR factorization + GMRES solve with stopping tolerance 10^{-9}

Matrix	n	Time (s)		Storage (GB)	
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-8}$
audikw_1	1.0M	1163	69	5	10
Bump_2911	2.9M	–	282	34	56
Emilia_923	0.9M	304	63	7	12
Fault_639	0.6M	–	45	5	9
Ga41As41H72	0.3M	–	76	12	17
Hook_1498	1.5M	902	75	6	11
Si87H76	0.2M	–	62	10	14

Low-accuracy BLR solvers:

☹ are slower and less robust

☺ but require much less storage

Improved preconditioner: context

Objective

- Compute solution to linear system $Ax = b$
- $A \in \mathbb{R}^{n \times n}$ is ill conditioned

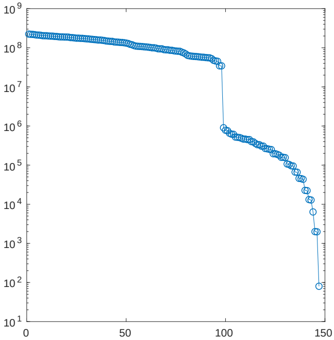
LU-based preconditioner

1. Compute approximate factorization $A = \widehat{L}\widehat{U} + \Delta A$
   - Half-precision factorization
   - Incomplete LU factorization
   - Structured matrix factorization: Block Low-Rank, $\mathcal{H}$, HSS,...
2. Solve $\Pi_{LU} Ax = \Pi_{LU} b$ with $\Pi_{LU} = \widehat{U}^{-1}\widehat{L}^{-1}$ via some iterative method

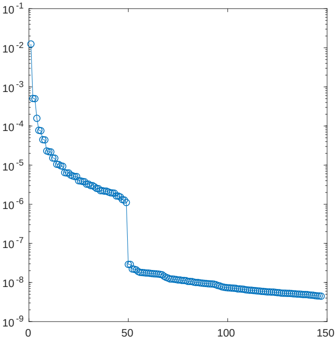
- Convergence to solution may be slow or fail

$\Rightarrow$ **Objective: accelerate convergence**

Matrix lund_a ($n = 147$, $\kappa(A) = 2.8e{+}06$)



SVD of A



SVD of A^{-1}

- Often, A is ill conditioned due to a small number of small singular values
- Then, A^{-1} is numerically low-rank

Improved preconditioner: key idea

Factorization error might be low-rank?

Let the error $E = \widehat{U}^{-1}\widehat{L}^{-1}A - I = \widehat{U}^{-1}\widehat{L}^{-1}(\widehat{L}\widehat{U} + \Delta A) - I$

$$= \widehat{U}^{-1}\widehat{L}^{-1}\Delta A \approx A^{-1}\Delta A$$

Does E retain the low-rank property of A^{-1}?

A novel preconditioner

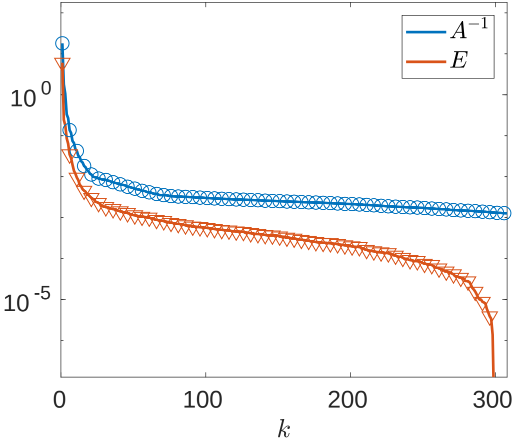
Consider the preconditioner

$$\Pi_{E_k} = (I + E_k)^{-1}\Pi_{LU}$$

with E_k a rank-k approximation to E.

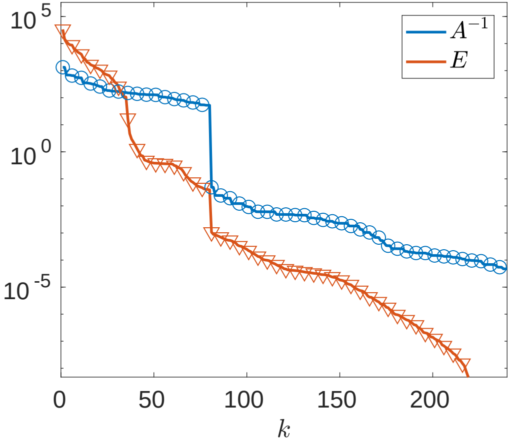
- If $E = E_k$, $\Pi_{E_k} = A^{-1}$
- If $E \approx E_k$ for some small k, Π_{E_k} can be computed cheaply

Typical SV distributions of A^{-1} and E



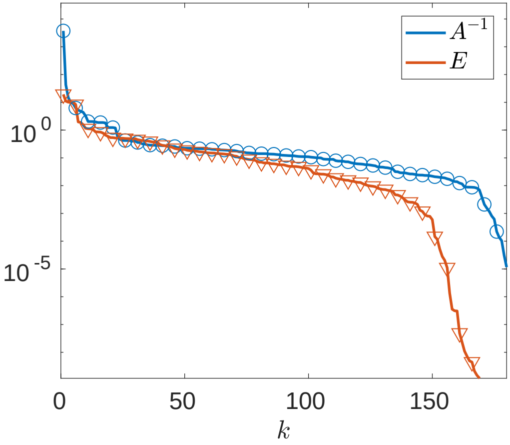
Matrix cz308

Typical SV distributions of A^{-1} and E

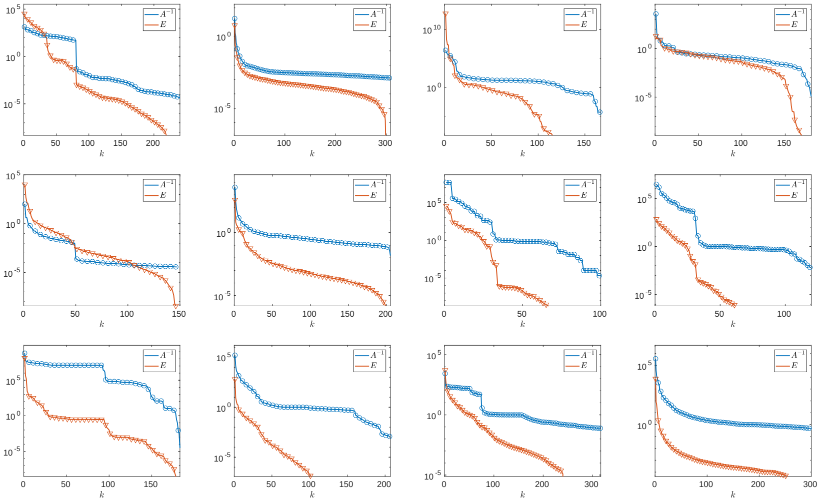


Matrix steam1

Typical SV distributions of A^{-1} and E



Matrix rajat14



We did **not** specifically select matrices for which A^{-1} is low-rank!

We need to compute a rank-k approximation of

$$E = \widehat{U}^{-1}\widehat{L}^{-1}A - I$$

E cannot be built explicitly! $\Rightarrow$ use **randomized** method

Algorithm 1 Randomized SVD via direct SVD of $V^T E$.

1: Sample E: $S = E\Omega$, with Ω a $n \times (k+p)$ random matrix.
2: Orthonormalize S: $V = \mathrm{qr}(S)$. $\{\Rightarrow E \approx VV^T E.\}$
3: Compute truncated SVD $V^T E \approx X_k \Sigma_k Y_k^T$.
4: $E_k \approx (VX_k)\Sigma_k Y_k^T$.

Improved BLR preconditioners

Results for $\varepsilon = 10^{-2}$:

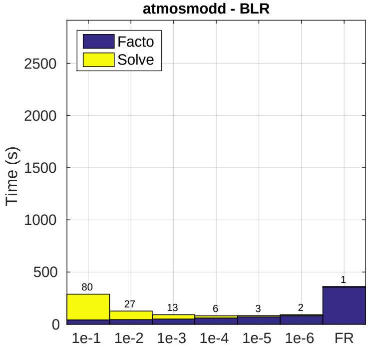
Matrix	Π_{LU}		Π_{E_k}	
	Iter.	Time	Iter.	Time
audikw_1	691	1163	331	625
Bump_2911	–	–	284	1708
Emilia_923	174	304	136	267
Fault_639	–	–	294	345
Ga41As41H72	–	–	135	143
Hook_1498	417	902	356	808
Si87H76	–	–	131	116

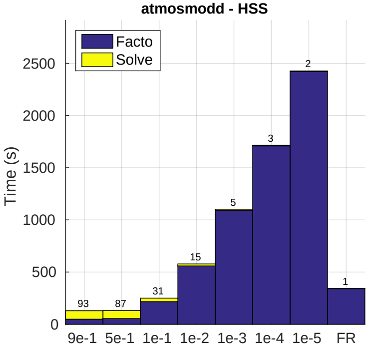
$\Rightarrow$ **performance and robustness improvement with zero storage overhead**

Comparison with a hierarchical solver

Comparison with STRUMPACK solver (HSS format):

C. Gorman, G. Chavez, P .Ghysels, T. Mary, F.-H. Rouet, and X. S. Li. *Matrix-free Construction of HSS Representation Using Adaptive Randomized Sampling*. Submitted (2018).





Comparatively with BLR, HSS favors low-accuracy preconditioning

Applying improved preconditioner to HSS should have an even greater impact!

Conclusion

Takeaway messages

Complexity

BLR factorization achieves quadratic dense complexity but (quasi-)linear sparse complexity with a small number of levels

Scalability

A large fraction of this theoretical reduction is converted into actual time gains, even on large numbers of cores

Stability

It is numerically stable thanks to numerical pivoting and can potentially exploit low-precision floating-point arithmetic

⇒ BLR solvers achieve a good compromise between asymptotic **complexity**, parallel **scalability**, and numerical **stability**

Perspectives

Complexity

- Can we exploit nested bases ($\mathcal{H}^2$, HSS) in BLR/MBLR?
- Exploiting fast matrix arithmetic (e.g. Strassen's algorithm)
- Asymptotic complexity of the solution phase with sparse RHS

Scalability

- MBLR performance and scalability analysis
- Distributed-memory: need for specialized scheduling strategies
- How to achieve good memory scalability?

Stability

- Rounding error analysis of multilevel and hierarchical solvers
- Exploiting half precision within low-rank preconditioners
- Numerical pivoting strategies for BLR factorization

References

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).

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).

P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*. ACM Trans. Math. Soft. (2018).

C. Gorman, G. Chavez, P .Ghysels, T. Mary, F.-H. Rouet, and X. S. Li. *Matrix-free Construction of HSS Representation Using Adaptive Randomized Sampling*. Submitted (2018).

N. Higham and T. Mary. *A New Preconditioner that Exploits Low-Rank Approximations to Factorization Error*. SIAM J. Sci. Comp (2018).

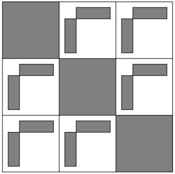
N. Higham and T. Mary. *A New Approach to Probabilistic Rounding Error Analysis*. Submitted (2018).

P. Amestoy, R. Brossier, A. Buttari, J.-Y. L'Excellent, T. Mary, L. Métivier, A. Miniussi, and S. Operto. *Fast 3D frequency-domain full waveform inversion with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea*. Geophysics (2016).

D. Shantsev, P. Jaysaval, S. Kethulle de Ryhove, P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. *Large-scale 3D EM modeling with a Block Low-Rank multifrontal direct solver*. Geophys. J. Int (2017).

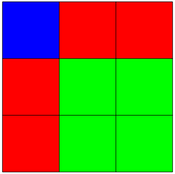
Backup slides

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



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

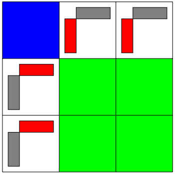
Assume all off-diagonal blocks are low-rank. Then:



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

$$
\begin{aligned}
FlopLU &= cost_{getrf} * nb_{getrf} + cost_{trsm} * nb_{trsm} + cost_{gemm} * nb_{gemm} \\
&= O(b^3) * O(\frac{m}{b}) + O(b^3) * O((\frac{m}{b})^2) + O(br^2) * O((\frac{m}{b})^3) \\
&= O(mb^2 + m^2 b + m^3 r^2/b^2) \\
&= \mathbf{O(m^{7/3} r^{2/3})} \text{ for } b = (mr^2)^{1/3}
\end{aligned}
$$

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



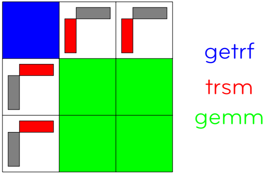
getrf
trsm
gemm

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

$$
\begin{aligned}
FlopLU &= cost_{getrf} * nb_{getrf} + cost_{trsm} * nb_{trsm} + cost_{gemm} * nb_{gemm} \\
&= O(b^3) * O(\frac{m}{b}) + O(\cancel{b^3} b^2 r) * O((\frac{m}{b})^2) + O(br^2) * O((\frac{m}{b})^3) \\
&= O(mb^2 + m^2 \cancel{b} r + m^3 r^2 / b^2) \\
&= \mathbf{O}(\cancel{m^{7/3} r^{2/3}} \mathbf{m^2 r}) \text{ for } b = \cancel{(mr^2)^{1/3}} (mr)^{1/2}
\end{aligned}
$$

CFU variant improves asymptotic complexity!

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



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

$$\begin{aligned} FlopLU &= cost_{getrf} * nb_{getrf} + cost_{trsm} * nb_{trsm} + cost_{gemm} * nb_{gemm} \\ &= O(b^3) * O(\frac{m}{b}) + O(\cancel{b^3} b^2 r) * O((\frac{m}{b})^2) + O(br^2) * O((\frac{m}{b})^3) \\ &= O(mb^2 + m^2 \cancel{b} r + m^3 r^2 / b^2) \\ &= \mathbf{O}(\cancel{m^{7/3} r^{2/3}} \mathbf{m^2 r}) \text{ for } b = \cancel{(mr^2)^{1/3}} (mr)^{1/2} \end{aligned}$$

CFU variant improves asymptotic complexity!

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

Nested dissection complexity formulas

In the **2D** case:

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

Nested dissection complexity formulas

In the **2D** case:

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

If $\mathcal{C}_{dense} = O(m^{\alpha})$, $\mathcal{C}_{sparse}$ is a geom. series of common ratio $2^{2-\alpha}$:

$$\mathcal{C}_{sparse} = \begin{cases} O(n^{\alpha/2}) & \text{if } \alpha > 2 \\ O(n \log n) & \text{if } \alpha = 2 \\ O(n) & \text{if } \alpha < 2 \end{cases}$$

Nested dissection complexity formulas

In the **2D** case:

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

If $\mathcal{C}_{dense} = O(m^{\alpha})$, $\mathcal{C}_{sparse}$ is a geom. series of common ratio $2^{2-\alpha}$:

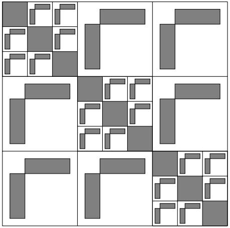
$$\mathcal{C}_{sparse} = \begin{cases} O(n^{\alpha/2}) & \text{if } \alpha > 2 \\ O(n \log n) & \text{if } \alpha = 2 \\ O(n) & \text{if } \alpha < 2 \end{cases}$$

Similar formulas in the **3D** case:

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

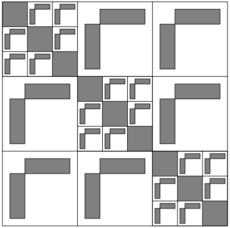
$$\mathcal{C}_{sparse} = \begin{cases} O(n^{2\alpha/3}) & \text{if } \alpha > 1.5 \\ O(n \log n) & \text{if } \alpha = 1.5 \\ O(n) & \text{if } \alpha < 1.5 \end{cases}$$

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



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

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



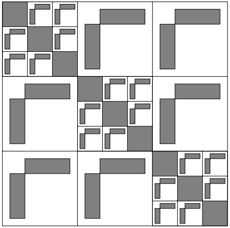
$$Storage = cost_{LR} * nb_{LR} + cost_{BLR} * nb_{BLR}$$
$$= O(br) * O((\frac{m}{b})^2) + O(b^{3/2} r^{1/2}) * O(\frac{m}{b})$$
$$= O(m^2 r/b + m(br)^{1/2})$$
$$= \mathbf{O(m^{4/3} r^{2/3})} \text{ for } b = (m^2 r)^{1/3}$$

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:



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

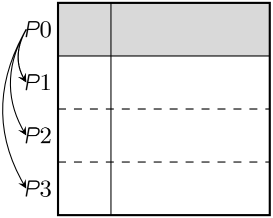
Similarly, we can prove:

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

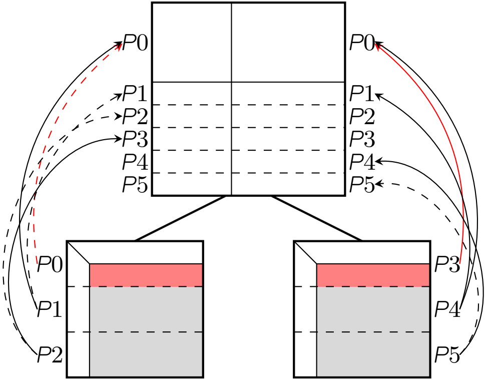
Result holds if a **constant** number of off-diag. blocks is BLR.

		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	$O(n^2)$	$O(n^{1.33})$	$O(n^{1.11})$	...	$O(n)$

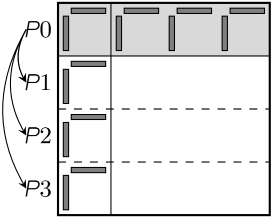
Type of messages



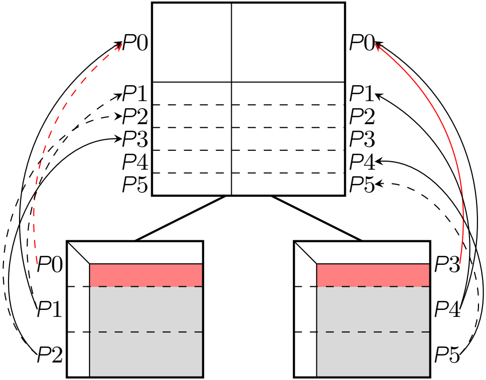
LU messages



CB messages



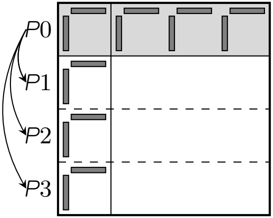
LU messages



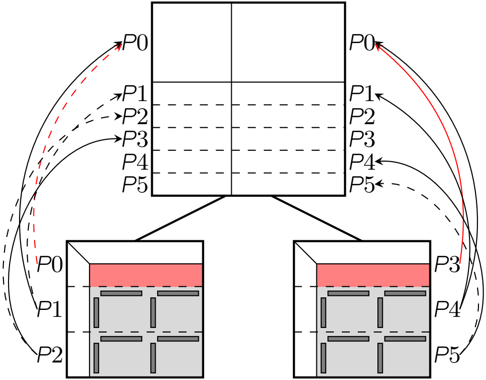
CB messages

- Volume of *LU* messages is reduced by compressing the factors
  - ☺ Reduces operation count, communications, and memory consumption

Type of messages

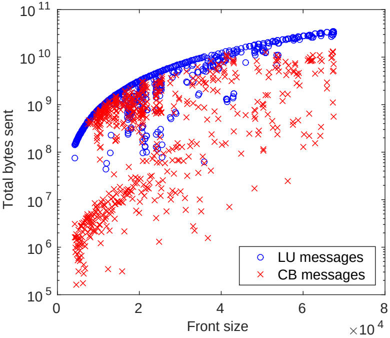


LU messages



CB messages

- Volume of *LU* messages is reduced by compressing the factors
  - ☺ Reduces operation count, communications, and memory consumption
- Volume of CB messages can be reduced by compressing the CB
  - ☺ Reduces communications and memory consumption
  - ☹ Increases operation count unless assembly is done in LR

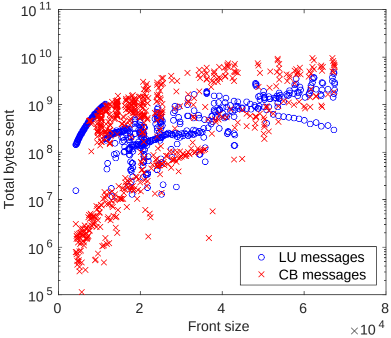


- FR case: *LU* messages dominate

Theoretical communication bounds

	$\mathcal{W}_{LU}$	$\mathcal{W}_{CB}$	$\mathcal{W}_{tot}$
FR	$\mathcal{O}\left(n^{4/3}p\right)$	$\mathcal{O}\left(n^{4/3}\right)$	$\mathcal{O}\left(n^{4/3}p\right)$

Communication analysis

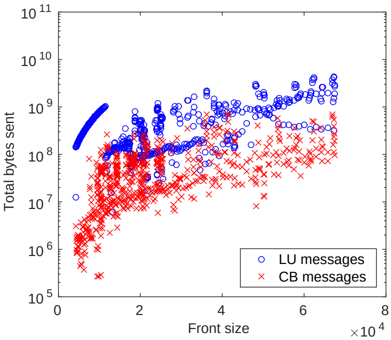


- FR case: *LU* messages dominate
- BLR case: CB messages dominate $\Rightarrow$ underwhelming reduction of communications

Theoretical communication bounds

	$\mathcal{W}_{LU}$	$\mathcal{W}_{CB}$	$\mathcal{W}_{tot}$
FR	$\mathcal{O}\left(n^{4/3}p\right)$	$\mathcal{O}\left(n^{4/3}\right)$	$\mathcal{O}\left(n^{4/3}p\right)$
BLR (CB_{FR})	$\mathcal{O}\left(nr^{1/2}p\right)$	$\mathcal{O}\left(n^{4/3}\right)$	$\mathcal{O}\left(nr^{1/2}p + n^{4/3}\right)$

Communication analysis



- FR case: *LU* messages dominate
- BLR case: CB messages dominate ⇒ underwhelming reduction of communications

⇒ CB compression allows for truly reducing the communications

Theoretical communication bounds

	$\mathcal{W}_{LU}$	$\mathcal{W}_{CB}$	$\mathcal{W}_{tot}$
FR	$\mathcal{O}\left(n^{4/3}p\right)$	$\mathcal{O}\left(n^{4/3}\right)$	$\mathcal{O}\left(n^{4/3}p\right)$
BLR (CB_{FR})	$\mathcal{O}\left(nr^{1/2}p\right)$	$\mathcal{O}\left(n^{4/3}\right)$	$\mathcal{O}\left(nr^{1/2}p + n^{4/3}\right)$
BLR (CB_{LR})	$\mathcal{O}\left(nr^{1/2}p\right)$	$\mathcal{O}\left(nr^{1/2}\right)$	$\mathcal{O}\left(nr^{1/2}p\right)$

Performance impact of the CB compression

matrix	10Hz	15Hz
order	17 M	58 M
factor flops (FR)	2.6 PF	29.6 PF
$\Rightarrow$ BLR (CB_{FR})	0.1 PF (5.3%)	1.0 PF (3.3%)
$\Rightarrow$ BLR (CB_{LR})	0.2 PF (6.1%)	1.1 PF (3.7%)
CB_{LR} flops impact	+15%	+12%
factor time (FR)	601	5,206
$\Rightarrow$ BLR (CB_{FR})	123 (4.9)	838 (6.2)
$\Rightarrow$ BLR (CB_{LR})	213 (2.8)	856 (6.1)
CB_{LR} time impact	+73%	+2%
comm. volume (FR)	5.3 TB	29.6 TB
comm. volume (CB_{FR})	1.7 TB (3.2)	13.3 TB (2.2)
comm. volume (CB_{LR})	0.6 TB (9.1)	1.2 TB (23.2)

$\Rightarrow$ CB compression becomes increasingly critical?

Performance impact of the CB compression

matrix	10Hz	15Hz
order	17 M	58 M
factor flops (FR)	2.6 PF	29.6 PF
$\Rightarrow$ BLR (CB_{FR})	0.1 PF (5.3%)	1.0 PF (3.3%)
$\Rightarrow$ BLR (CB_{LR})	0.2 PF (6.1%)	1.1 PF (3.7%)
CB_{LR} flops impact	+15%	+12%
factor time (FR)	601	5,206
$\Rightarrow$ BLR (CB_{FR})	123 (4.9)	838 (6.2)
$\Rightarrow$ BLR (CB_{LR})	213 (2.8)	856 (6.1)
CB_{LR} time impact	+73%	+2%
comm. volume (FR)	5.3 TB	29.6 TB
comm. volume (CB_{FR})	1.7 TB (3.2)	13.3 TB (2.2)
comm. volume (CB_{LR})	0.6 TB (9.1)	1.2 TB (23.2)

$\Rightarrow$ CB compression becomes increasingly critical?

Performance impact of the CB compression

matrix	10Hz	15Hz
order	17 M	58 M
factor flops (FR)	2.6 PF	29.6 PF
⇒ BLR (CB_{FR})	0.1 PF (5.3%)	1.0 PF (3.3%)
⇒ BLR (CB_{LR})	0.2 PF (6.1%)	1.1 PF (3.7%)
CB_{LR} flops impact	+15%	+12%
factor time (FR)	601	5,206
⇒ BLR (CB_{FR})	123 (4.9)	838 (6.2)
⇒ BLR (CB_{LR})	213 (2.8)	856 (6.1)
CB_{LR} time impact	+73%	+2%
comm. volume (FR)	5.3 TB	29.6 TB
comm. volume (CB_{FR})	1.7 TB (3.2)	13.3 TB (2.2)
comm. volume (CB_{LR})	0.6 TB (9.1)	1.2 TB (23.2)

⇒ CB compression becomes increasingly critical?

Some ingredients for the proof

The proof is based on *Stability of Block Algorithms with Fast Level-3 BLAS* (Demmel and Higham, 1992)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Inductive proof: easy to bound error of computing $S = A_{22} - L_{21}U_{12}$ and error of $S = L_{22}U_{22}$ is obtained by induction

Some ingredients for the proof

The proof is based on *Stability of Block Algorithms with Fast Level-3 BLAS* (Demmel and Higham, 1992)

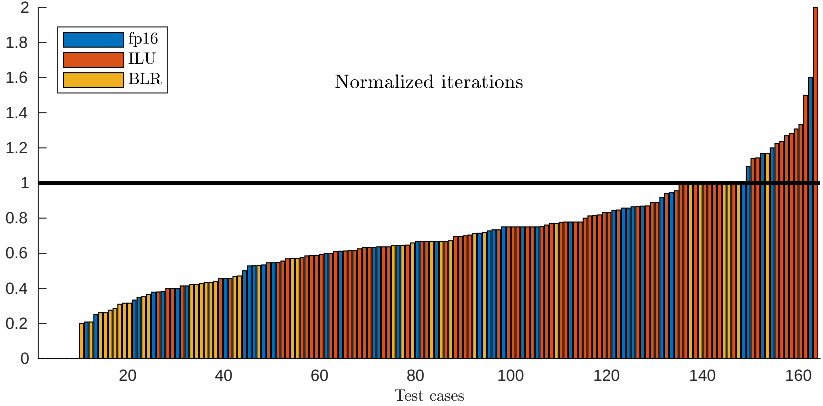
$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Inductive proof: easy to bound error of computing $S = A_{22} - L_{21}U_{12}$ and error of $S = L_{22}U_{22}$ is obtained by induction

For BLR, several specific difficulties arise:

- Need to bound error of low-rank product kernel: $C = \widetilde{A}\widetilde{B} = X_A \left(Y_A^T X_B\right) Y_B^T$
- Choice of norm matters: to obtain best constants possible, we need a consistent, unitarily invariant norm
- Global bound is obtained from blockwise bounds
  $\Rightarrow$ we work with the Frobenius norm

Black-box setting: use $p = 10$ and $k =$ num. rank at acc. 10^{-7}



Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices

$\Rightarrow$ but only needed after factorization

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices

$\Rightarrow$ but only needed after factorization

Traditional multifrontal storage is $S_A + S_{LU} + S_{CB}$

- S_A = storage for matrix A
- S_{LU} = storage for (BLR) LU factors
- S_{CB} = storage for contribution blocks $\Rightarrow$ temporary storage during factorization

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices

$\Rightarrow$ but only needed after factorization

Traditional multifrontal storage is $S_A + S_{LU} + S_{CB}$

- $S_A =$ storage for matrix A
- $S_{LU} =$ storage for (BLR) LU factors
- $S_{CB} =$ storage for contribution blocks $\Rightarrow$ temporary storage during factorization

Thus, S_{CB} and S_{E_k} do not overlap!

- Factorization storage: $S_A + S_{LU} + S_{CB}$
- Solution storage: $S_A + S_{LU} + S_{E_k}$

$\Rightarrow$ Total storage: $S_A + S_{LU} + \max(S_{CB}, S_{E_k})$

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices
$\Rightarrow$ but only needed after factorization

Traditional multifrontal storage is $S_A + S_{LU} + S_{CB}$

- S_A = storage for matrix A
- S_{LU} = storage for (BLR) LU factors
- S_{CB} = storage for contribution blocks $\Rightarrow$ temporary storage during factorization

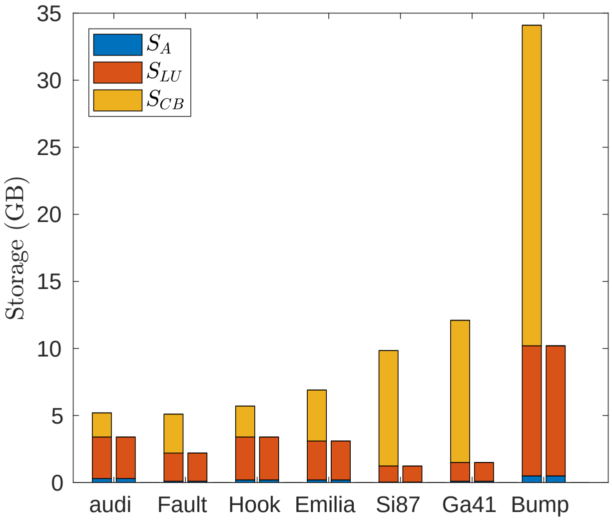
Thus, S_{CB} and S_{E_k} do not overlap!

- Factorization storage: $S_A + S_{LU} + S_{CB}$
- Solution storage: $S_A + S_{LU} + S_{E_k}$

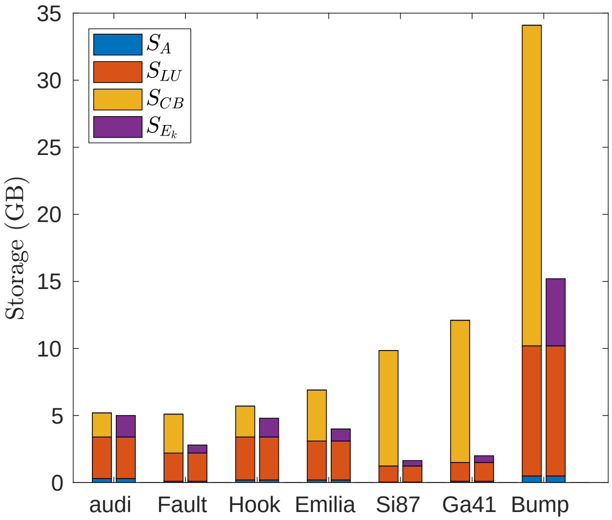
$\Rightarrow$ Total storage: $S_A + S_{LU} + \max(S_{CB}, S_{E_k})$

If $S_{E_k} \leq S_{CB}$, zero storage overhead!

Storage overhead: results



Storage overhead: results



$\Rightarrow$ **zero storage overhead on all matrices**