

Block Low-Rank Multifrontal Solvers: complexity, performance, and scalability

P. Amestoy^{*1} A. Buttari^{*,2} J.-Y. L'Excellent^{†,3} T. Mary^{*,4}

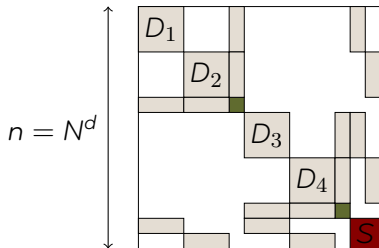
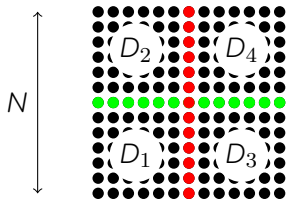
^{*}Université de Toulouse [†]ENS Lyon

¹INPT-IRIT ²CNRS-IRIT ³INRIA-LIP ⁴UPS-IRIT

Sparse Days, 6-8 Sep. 2017, Toulouse

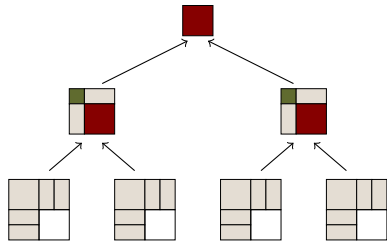
Introduction

Multifrontal Factorization with Nested Dissection



3D problem complexity

→ Flops: $O(n^2)$, mem: $O(n^{4/3})$



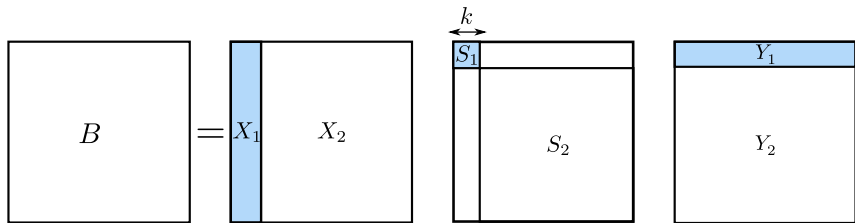
Low-rank matrices

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

$$\boxed{B} = \boxed{X} \boxed{S} \boxed{Y}$$

Low-rank matrices

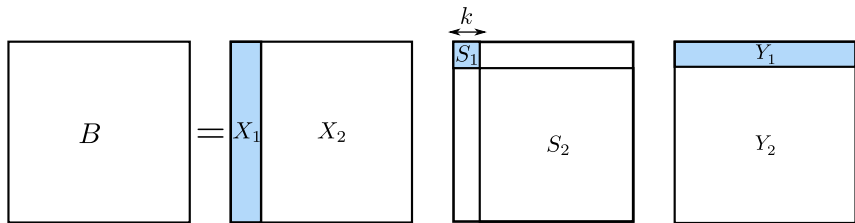
Take a dense matrix B of size $b \times b$ and compute its SVD $B = XSY$:



$$B = X_1 S_1 Y_1 + X_2 S_2 Y_2 \quad \text{with} \quad S_1(k, k) = \sigma_k > \varepsilon, \quad S_2(1, 1) = \sigma_{k+1} \leq \varepsilon$$

Low-rank matrices

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



$$B = X_1 S_1 Y_1 + X_2 S_2 Y_2 \quad \text{with} \quad S_1(k, k) = \sigma_k > \varepsilon, \quad S_2(1, 1) = \sigma_{k+1} \leq \varepsilon$$

$$\text{If } \tilde{B} = X_1 S_1 Y_1 \quad \text{then} \quad \|B - \tilde{B}\|_2 = \|X_2 S_2 Y_2\|_2 = \sigma_{k+1} \leq \varepsilon$$

Low-rank matrices

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



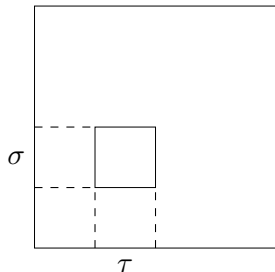
$$B = X_1 S_1 Y_1 + X_2 S_2 Y_2 \quad \text{with} \quad S_1(k, k) = \sigma_k > \varepsilon, \quad S_2(1, 1) = \sigma_{k+1} \leq \varepsilon$$

$$\text{If } \tilde{B} = X_1 S_1 Y_1 \quad \text{then} \quad \|B - \tilde{B}\|_2 = \|X_2 S_2 Y_2\|_2 = \sigma_{k+1} \leq \varepsilon$$

If the singular values of B decay very fast (e.g. exponentially) then $k \ll b$ even for very small ε (e.g. 10^{-14}) $\Rightarrow$ memory and CPU consumption can be reduced considerably with a controlled loss of accuracy ($\leq \varepsilon$) if $\tilde{B}$ is used instead of B

Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**

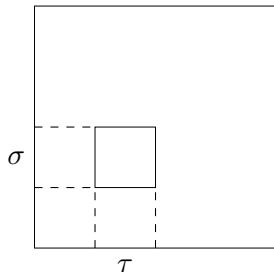


A block B represents the interaction between two subdomains σ and τ .

If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.

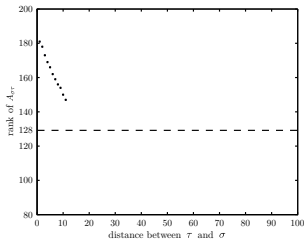
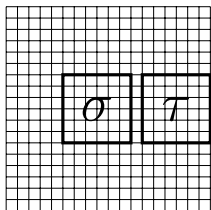
Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**



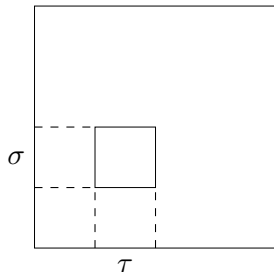
A block B represents the interaction between two subdomains σ and τ .

If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.

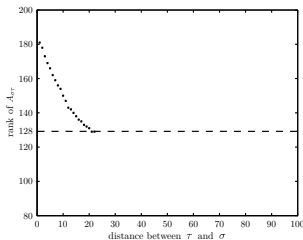
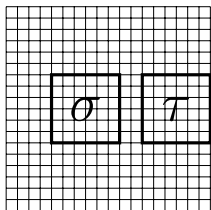


Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**

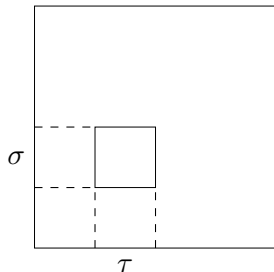


A block B represents the interaction between two subdomains σ and τ .
If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.

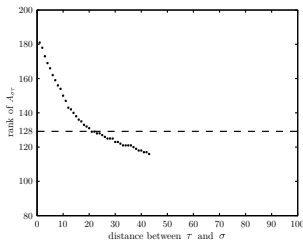
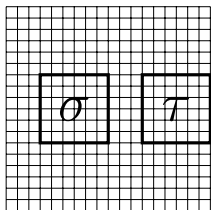


Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**

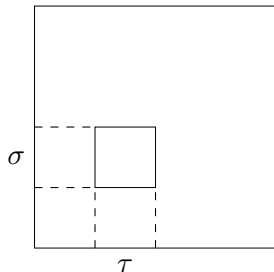


A block B represents the interaction between two subdomains σ and τ .
If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.



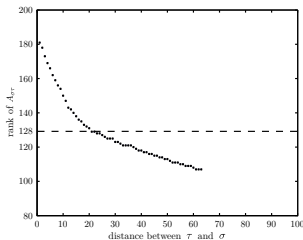
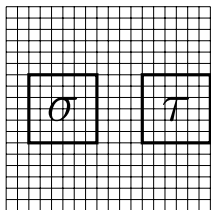
Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**



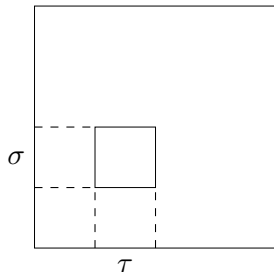
A block B represents the interaction between two subdomains σ and τ .

If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.



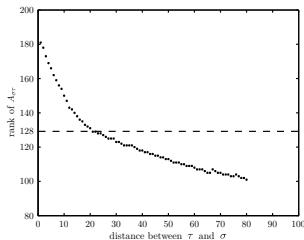
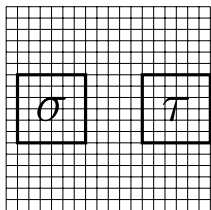
Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**



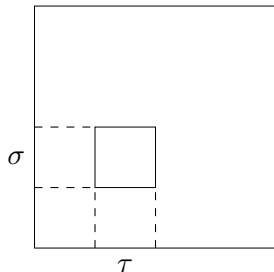
A block B represents the interaction between two subdomains σ and τ .

If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.



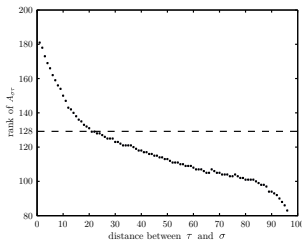
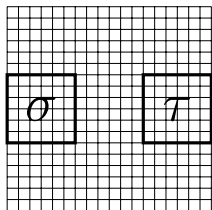
Low-rank matrix formats

Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**

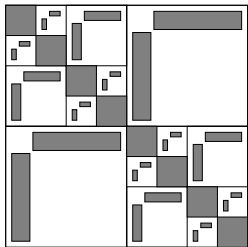


A block B represents the interaction between two subdomains σ and τ .

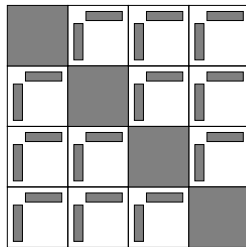
If they have a **small diameter** and are **far away** their interaction is weak $\Rightarrow$ rank is low.



$\mathcal{H}$ and BLR matrices

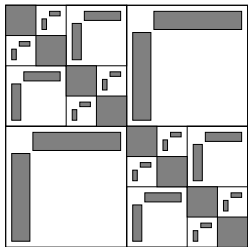


$\mathcal{H}$ -matrix

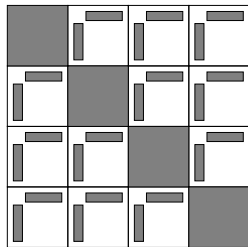


BLR matrix

$\mathcal{H}$ and BLR matrices



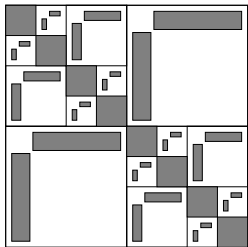
$\mathcal{H}$ -matrix



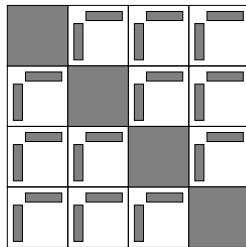
BLR matrix

- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure
- Theoretical complexity?
 $\Rightarrow O(n^{4/3})$, as we will prove
- Simple structure

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$ -matrix

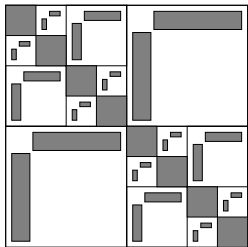


BLR matrix

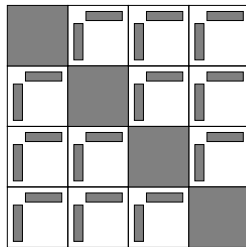
- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure
- Theoretical complexity?
 $\Rightarrow O(n^{4/3})$, as we will prove
- Simple structure

Find a good compromise between complexity and performance

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$ -matrix



BLR matrix

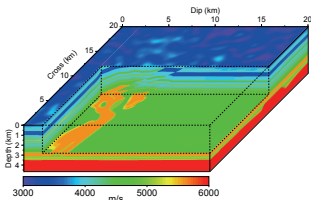
- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure
- Theoretical complexity?
 $\Rightarrow O(n^{4/3})$, as we will prove
- Simple structure

Find a good compromise between complexity and performance

$\Rightarrow$ Ongoing collaboration with STRUMPACK team (LBNL) to compare BLR and hierarchical formats

Applications

Experimental Setting: Matrices (1/3)



3D Seismic Modeling

Helmholtz equation

Single complex (c) arithmetic

Unsymmetric LU factorization

Required accuracy: $\varepsilon = 10^{-3}$

Credits: SEISCOPE

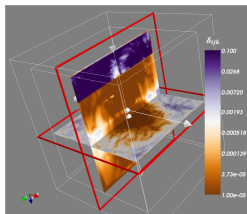
matrix	n	nnz	flops	storage
5Hz	2.9M	70M	65.0 TF	59.7 GB
7Hz	7.2M	177M	404.2 TF	205.0 GB
10Hz	17.2M	446M	2.6 PF	710.8 GB

Full-Rank statistics

- Amestoy, Brossier, Buttari, L'Excellent, Mary, Métivier, Miniussi, and 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.

Experimental Setting: Matrices (2/3)

E_{xx} , BLR STRATEGY 2, $IR = 0$, $\varepsilon_{BLR} = 10^{-7}$



Relative deviation between E_x -components of low-rank and full-rank solutions

$$\delta_{ijk} = \frac{|E_{ijk} - \tilde{E}_{ijk}|}{\sqrt{\frac{|E_{ijk}|^2 + |\tilde{E}_{ijk}|^2}{2}} + \epsilon}$$

(only for E_x)

emgs

3D Electromagnetic Modeling

Maxwell equation

Double complex (z) arithmetic

Symmetric LDL^T factorization

Required accuracy: $\varepsilon = 10^{-7}$

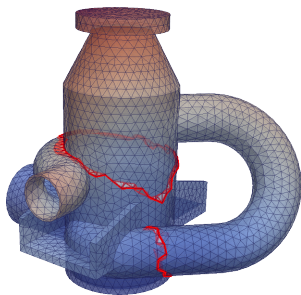
Credits: EMGS

matrix	n	nnz	flops	storage
E3	2.9M	37M	57.9 TF	77.5 GB
E4	17M	226M	1.8 PF	1.7 TB
S3	3.3M	43M	78.0 TF	94.6 GB
S4	21M	266M	2.5 PF	2.1 TB

Full-Rank statistics

- Shantsev, Jaysaval, de la Kethulle de Ryhove, Amestoy, Buttari, L'Excellent, and Mary. *Large-scale 3D EM modeling with a Block Low-Rank multifrontal direct solver*, Geophysical Journal International, 2017.

Sparse Days, 6-8 Sep. 2017, Toulouse



3D Structural Mechanics

Double real (d) arithmetic

Symmetric LDL^T factorization

Required accuracy: $\varepsilon = 10^{-9}$

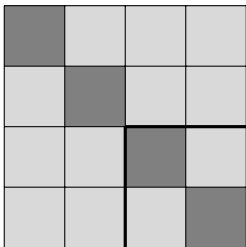
Credits: Code_Aster (EDF)

matrix	n	nnz	flops	storage
perf008d	1.9M	81M	101.0 TF	52.6 GB
perf008ar	3.9M	159M	377.5 TF	129.8 GB
perf008cr	7.9M	321M	1.6 PF	341.1 GB
perf009ar	5.4M	209M	23.4 TF	40.2 GB

Full-Rank statistics

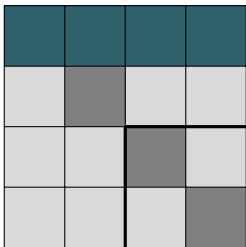
The Block-Low Rank Factorization

Standard BLR factorization: FSCU



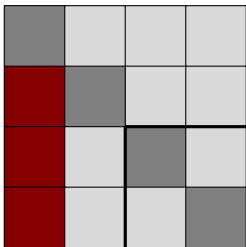
- FSCU

Standard BLR factorization: FSCU



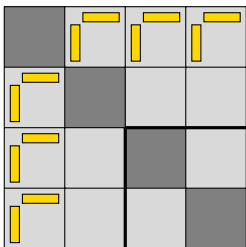
- FSCU (Factor,

Standard BLR factorization: FSCU



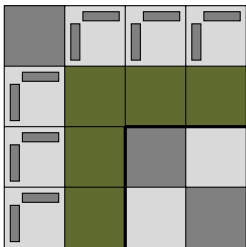
- FSCU (Factor, Solve,

Standard BLR factorization: FSCU



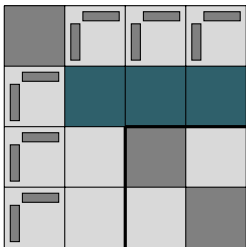
- FSCU (Factor, Solve, Compress,

Standard BLR factorization: FSCU



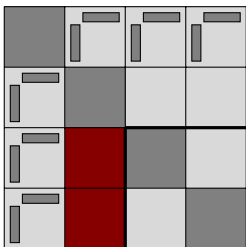
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



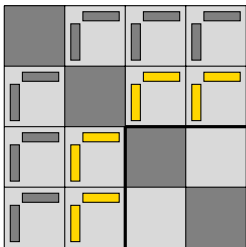
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



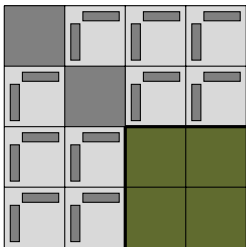
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



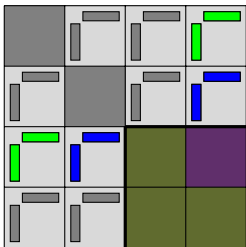
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



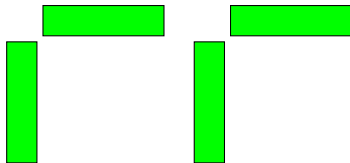
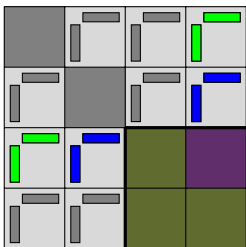
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



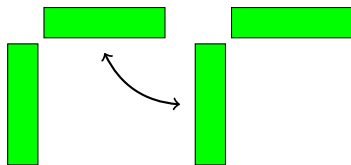
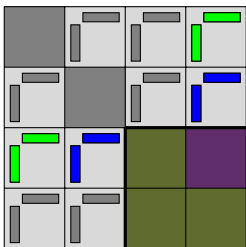
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



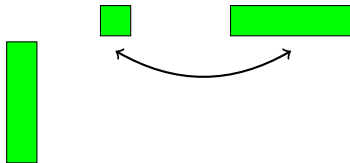
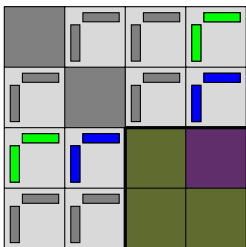
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



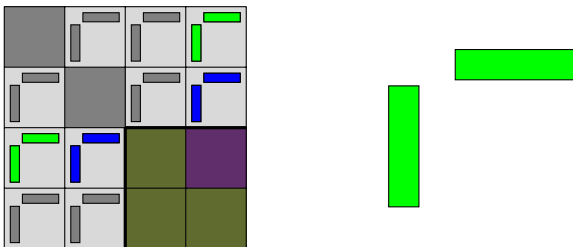
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



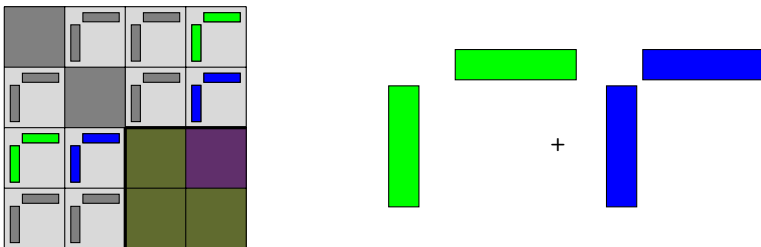
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



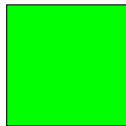
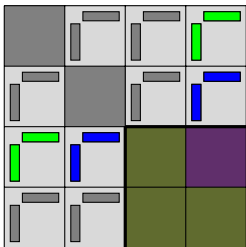
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU

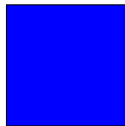


- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU

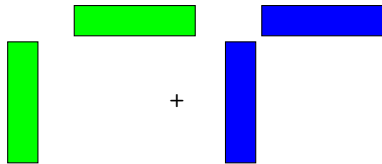
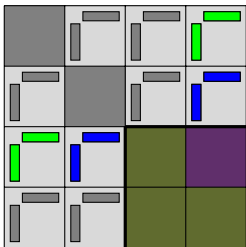


+



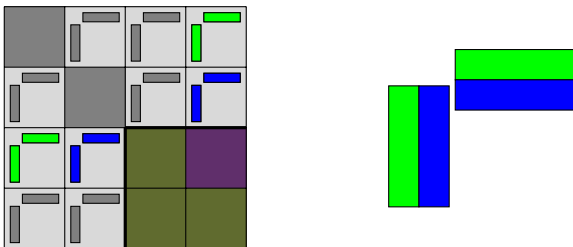
- FSCU (Factor, Solve, Compress, Update)

LUAR variant: accumulation and recompression



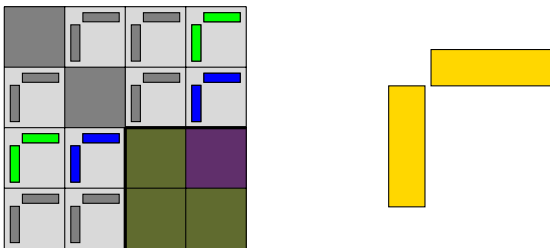
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR

LUAR variant: accumulation and recompression



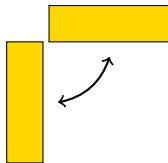
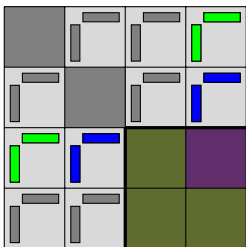
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations

LUAR variant: accumulation and recompression



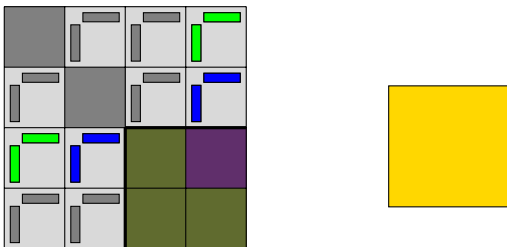
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with LSTC to design efficient recompression strategies

LUAR variant: accumulation and recompression

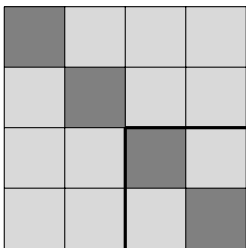


- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with **LSTC** to design efficient recompression strategies

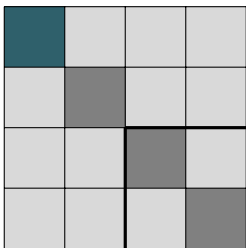
LUAR variant: accumulation and recompression



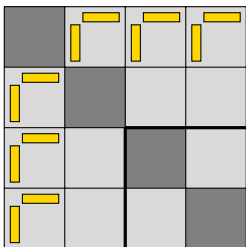
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ **complexity reduction**: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with **LSTC** to design efficient recompression strategies



- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with LSTC to design efficient recompression strategies
- FCSU(+LUAR)

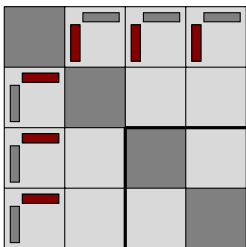


- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with LSTC to design efficient recompression strategies
- FCSU(+LUAR)
 - Restricted pivoting, e.g. to diagonal blocks



- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ **complexity reduction**: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with **LSTC** to design efficient recompression strategies
- FCSU(+LUAR)
 - Restricted pivoting, e.g. to diagonal blocks

FCSU variant: compress before solve



- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with LSTC to design efficient recompression strategies
- FCSU(+LUAR)
 - Restricted pivoting, e.g. to diagonal blocks
 - Low-rank Solve $\Rightarrow$ complexity reduction: $O(n^{\frac{11}{6}}) \rightarrow O(n^{\frac{4}{3}})$

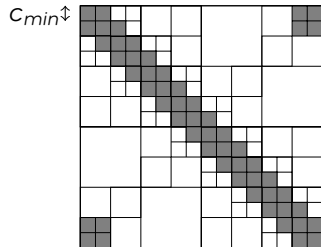
Complexity of the factorization

Until recently, BLR complexity was unknown.

Can we use $\mathcal{H}$ theory on BLR matrices?

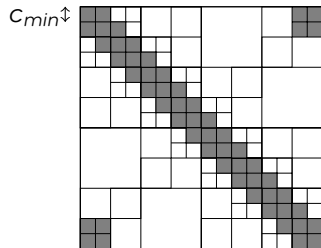
Until recently, BLR complexity was unknown.

Can we use $\mathcal{H}$ theory on BLR matrices?



Until recently, BLR complexity was unknown.

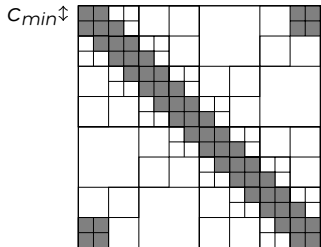
Can we use $\mathcal{H}$ theory on BLR matrices?



Complexity mainly depends on r_{max} ,
the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

Until recently, BLR complexity was unknown.

Can we use $\mathcal{H}$ theory on BLR matrices?

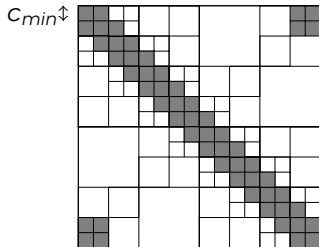


Complexity mainly depends on r_{max} ,
the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

- **Problem:** in $\mathcal{H}$ formalism, the maxrank of the blocks of a BLR matrix is $r_{max} = b$ (due to full-rank blocks)

Until recently, BLR complexity was unknown.

Can we use $\mathcal{H}$ theory on BLR matrices?

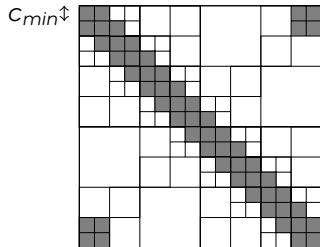


Complexity mainly depends on r_{max} ,
the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

- **Problem:** in $\mathcal{H}$ formalism, the maxrank of the blocks of a BLR matrix is $r_{max} = b$ (due to full-rank blocks)
- $\mathcal{H}$ theory applied to BLR does not give a satisfying result

Until recently, BLR complexity was unknown.

Can we use $\mathcal{H}$ theory on BLR matrices?



Complexity mainly depends on r_{max} ,
the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

- **Problem:** in $\mathcal{H}$ formalism, the maxrank of the blocks of a BLR matrix is $r_{max} = b$ (due to full-rank blocks)
- $\mathcal{H}$ theory applied to BLR does not give a satisfying result
- **Solution:** extend the theory by bounding the number of full-rank blocks

► Amestoy, Buttari, L'Excellent, and Mary. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, SIAM SISC, 2016.

Complexity of multifrontal BLR factorization

	operations (OPC)		factor size (NNZ)	
	$r = O(1)$	$r = O(N)$	$r = O(1)$	$r = O(N)$
FR	$O(n^2)$	$O(n^2)$	$O(n^{\frac{4}{3}})$	$O(n^{\frac{4}{3}})$
BLR	$O(n^{\frac{4}{3}}) - O(n^{\frac{5}{3}})$	$O(n^{\frac{5}{3}}) - O(n^{\frac{11}{6}})$	$O(n \log n)$	$O(n^{\frac{7}{6}} \log n)$
$\mathcal{H}$	$O(n \log n)$	$O(n^{\frac{4}{3}} \log n)$	$O(n \log n)$	$O(n^{\frac{7}{6}} \log n)$

in the 3D case (similar analysis possible for 2D)

Important properties: with both $r = O(1)$ or $r = O(N)$

- Complexity depends on how the BLR factorization is performed
- The BLR complexity exponent is always lower than the FR one
- The best BLR complexity is not so far from the $\mathcal{H}$ -case

Complexity of multifrontal BLR factorization

	operations (OPC)		factor size (NNZ)	
	$r = O(1)$	$r = O(N)$	$r = O(1)$	$r = O(N)$
FR	$O(n^2)$	$O(n^2)$	$O(n^{\frac{4}{3}})$	$O(n^{\frac{4}{3}})$
BLR	$O(n^{\frac{4}{3}}) - O(n^{\frac{5}{3}})$	$O(n^{\frac{5}{3}}) - O(n^{\frac{11}{6}})$	$O(n \log n)$	$O(n^{\frac{7}{6}} \log n)$
$\mathcal{H}$	$O(n \log n)$	$O(n^{\frac{4}{3}} \log n)$	$O(n \log n)$	$O(n^{\frac{7}{6}} \log n)$

in the 3D case (similar analysis possible for 2D)

Important properties: with both $r = O(1)$ or $r = O(N)$

- Complexity depends on how the BLR factorization is performed
- The BLR complexity exponent is always lower than the FR one
- The best BLR complexity is not so far from the $\mathcal{H}$ -case

How to convert complexity reduction into performance gain?

Performance on Multicores

Experiments are done on the **brunch** shared-memory machine of the LIP laboratory of Lyon:

- Four Intel(r) 24-cores Broadwell @ **2,2 GHz**
- Peak per core is **35.2 GF/s**
- Total memory is **1.5 TB**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**
 - Multithreaded (24 threads): **1.7 ratio**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**
 - Multithreaded (24 threads): **1.7 ratio**
- **Tree-based multithreading** is critical because the bottom of the assembly tree has a higher relative cost in BLR $\Rightarrow$ **1.9 ratio**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**
 - Multithreaded (24 threads): **1.7 ratio**
- **Tree-based multithreading** is critical because the bottom of the assembly tree has a higher relative cost in BLR $\Rightarrow$ **1.9 ratio**
- **Left-looking factorization** reduces the volume of memory transfer in BLR ("communication-avoiding") $\Rightarrow$ **2.4 ratio**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**
 - Multithreaded (24 threads): **1.7 ratio**
- **Tree-based multithreading** is critical because the bottom of the assembly tree has a higher relative cost in BLR $\Rightarrow$ **1.9 ratio**
- **Left-looking factorization** reduces the volume of memory transfer in BLR ("communication-avoiding") $\Rightarrow$ **2.4 ratio**
- **Accumulation (LUA)** $\Rightarrow$ **2.5 ratio**

Getting Gflops/s out of the BLR factorization

Follow the **FR/BLR ratio** on matrix S3

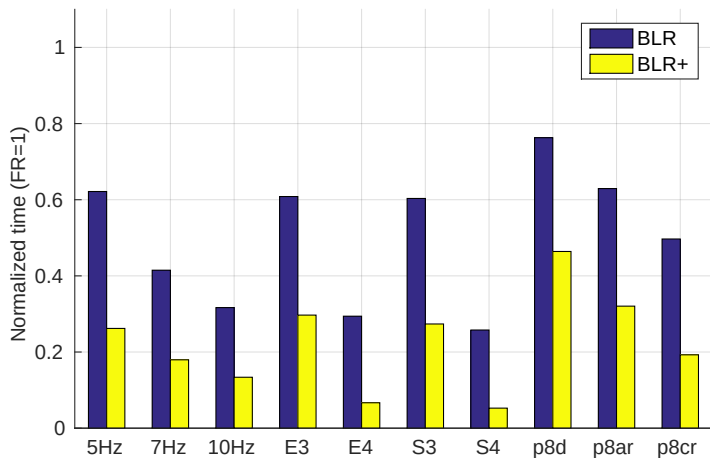
- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**
 - Multithreaded (24 threads): **1.7 ratio**
- **Tree-based multithreading** is critical because the bottom of the assembly tree has a higher relative cost in BLR $\Rightarrow$ **1.9 ratio**
- **Left-looking factorization** reduces the volume of memory transfer in BLR ("communication-avoiding") $\Rightarrow$ **2.4 ratio**
- **Accumulation** (LUA) $\Rightarrow$ **2.5 ratio**
- **Recompression** (LUAR) $\Rightarrow$ **2.6 ratio**

Getting Gflops/s out of the BLR factorization

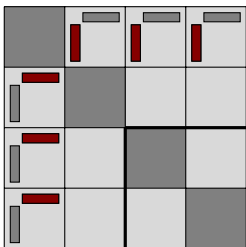
Follow the **FR/BLR ratio** on matrix S3

- Flop: **7.7 ratio**
- Time:
 - Sequential (1 thread): **3.3 ratio**
 - Multithreaded (24 threads): **1.7 ratio**
- **Tree-based multithreading** is critical because the bottom of the assembly tree has a higher relative cost in BLR $\Rightarrow$ **1.9 ratio**
- **Left-looking factorization** reduces the volume of memory transfer in BLR ("communication-avoiding") $\Rightarrow$ **2.4 ratio**
- **Accumulation (LUA)** $\Rightarrow$ **2.5 ratio**
- **Recompression (LUAR)** $\Rightarrow$ **2.6 ratio**
- **Compress before Solve (FCSU)** $\Rightarrow$ **3.6 ratio**

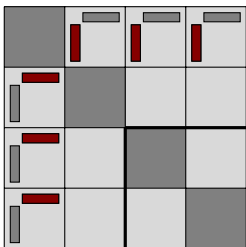
Multicore performance results (24 threads)



- Amestoy, Buttari, L'Excellent, and Mary. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*, submitted to ACM TOMS, 2017.

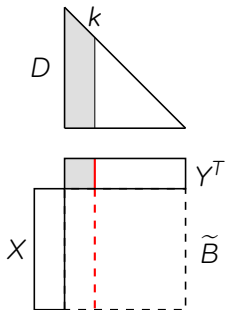


- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with LSTC to design efficient recompression strategies
- FCSU(+LUAR)
 - Restricted pivoting, e.g. to diagonal blocks
 - Low-rank Solve $\Rightarrow$ complexity reduction: $O(n^{\frac{11}{6}}) \rightarrow O(n^{\frac{4}{3}})$



- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
 - Better granularity in Update operations
 - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
 $\Rightarrow$ Collaboration with LSTC to design efficient recompression strategies
- FCSU(+LUAR)
 - Restricted pivoting, e.g. to diagonal blocks $\Rightarrow$ **not acceptable in many applications**
 - Low-rank Solve $\Rightarrow$ complexity reduction: $O(n^{\frac{11}{6}}) \rightarrow O(n^{\frac{4}{3}})$

Compress before Solve + pivoting: CFSU variant



How to **assess the quality** of pivot k ?

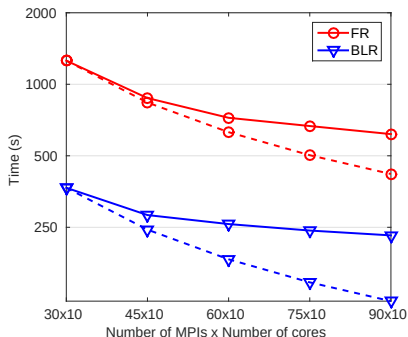
We need to estimate $\|\tilde{B}_{:,k}\|_{\max}$:

$\|\tilde{B}_{:,k}\|_{\max} \leq \|\tilde{B}_{:,k}\|_2 = \|XY_{k,:}^T\|_2 = \|Y_{k,:}^T\|_2$,
assuming X is orthonormal (e.g. RRQR, SVD).

matrix	residual			flops (% FR)		
	FSCU	FSCU	CFSU	FSCU	FSCU	CFSU
af_shell10	2e-06	5e-06	4e-06	29.9	22.7	22.7
Lin	4e-05	4e-05	4e-05	24.0	18.5	18.5
mario002	2e-06	fail	1e-06	82.8	—	72.2
perf009ar	3e-13	1e-01	9e-11	26.0	22.7	22.1

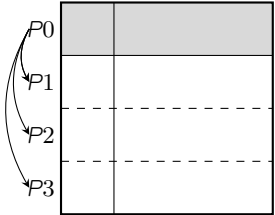
Distributed-memory BLR factorization

Strong scalability analysis

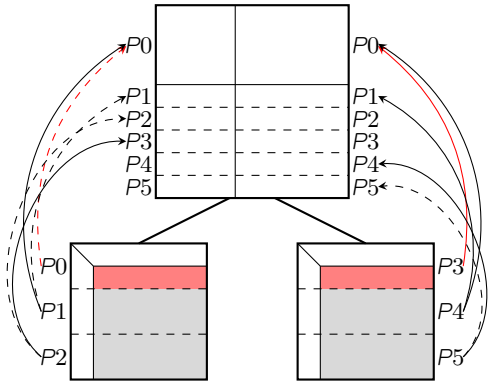


- Flops reduced by 12.8 but volume of communications only by 2.2 $\Rightarrow$ **higher relative weight of communications**
- Load unbalance (ratio between most and less loaded processes) increases from 1.28 to 2.57

Communication analysis

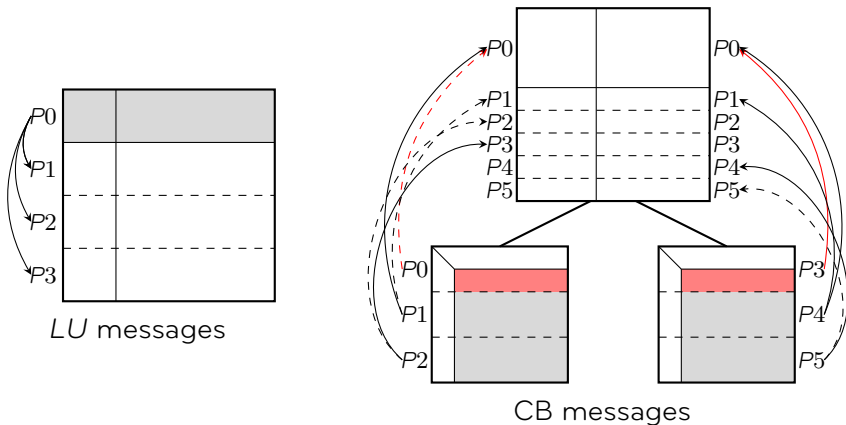


LU messages

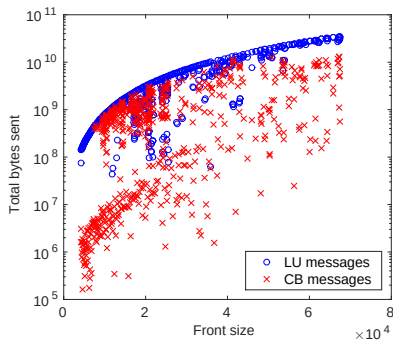


CB messages

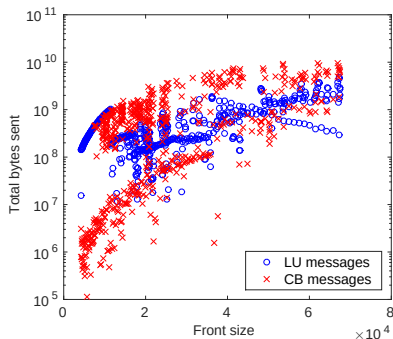
Communication analysis



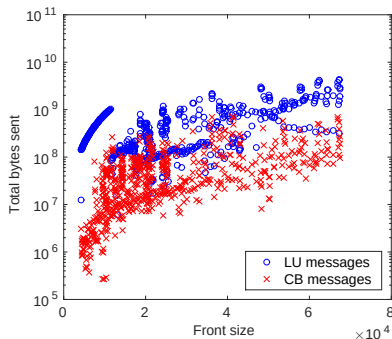
- Volume of *LU* messages is reduced in BLR (compressed factors)
- Volume of CB messages can be reduced by **compressing the CB** $\Rightarrow$ but it is an **overhead cost**



- FR case: *LU* messages dominate

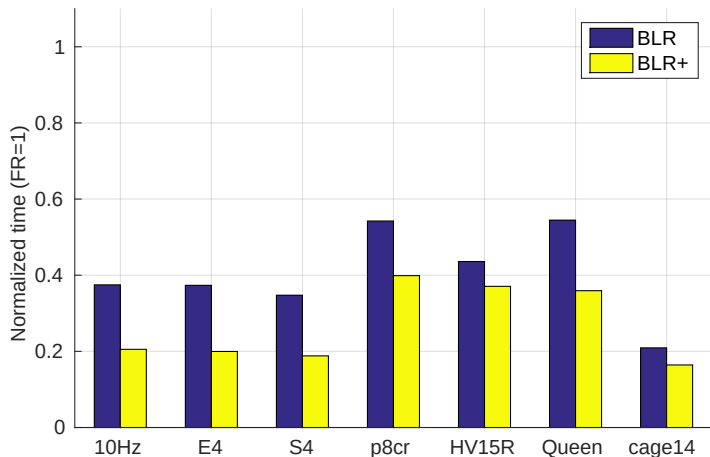


- FR case: *LU* messages dominate
- BLR case: CB messages dominate $\Rightarrow$ underwhelming reduction of comms.



- FR case: *LU* messages dominate
 - BLR case: CB messages dominate $\Rightarrow$ underwhelming reduction of comms.
- $\Rightarrow$ **CB compression** allows for truly reducing the comms. Represents an **overhead cost** but may lead to speedups depending on **network speed w.r.t. processor speed**

Distributed performance results (90×10 cores)



⇒ promising preliminary results, much work left to do!

Conclusion

References and acknowledgements

Software

- MUMPS 5.1.0

Publications

- ▶ Amestoy, Buttari, L'Excellent, and Mary. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, SIAM SISC, 2017.
- ▶ Amestoy, Buttari, L'Excellent, and Mary. *Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures*, submitted to ACM TOMS, 2017.
- ▶ Amestoy, Brossier, Buttari, L'Excellent, Mary, Métivier, Miniussi, and 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.
- ▶ Shantsev, Jaysaval, de la Kethulle de Ryhove, Amestoy, Buttari, L'Excellent, and Mary. *Large-scale 3D EM modeling with a Block Low-Rank multifrontal direct solver*, Geophysical Journal International, 2017.

Acknowledgements

- LIP and CALMIP for providing access to the machines
- EMGS, SEISCOPE, and EDF for providing the matrices



Thanks!
Questions?

Backup Slides

1. **Poisson:** N^3 grid with a 7-point stencil with $u = 1$ on the boundary $\partial\Omega$

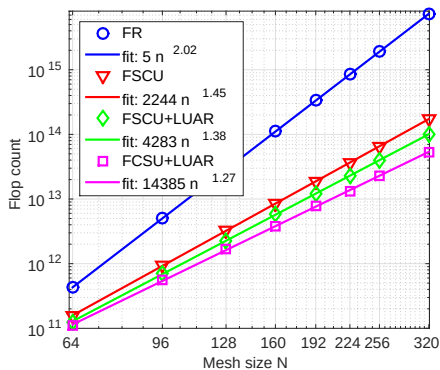
$$\Delta u = f$$

2. **Helmholtz:** N^3 grid with a 27-point stencil, ω is the angular frequency, $v(x)$ is the seismic velocity field, and $u(x, \omega)$ is the time-harmonic wavefield solution to the forcing term $s(x, \omega)$.

$$\left(-\Delta - \frac{\omega^2}{v(x)^2} \right) u(x, \omega) = s(x, \omega)$$

ω is fixed and equal to 4Hz.

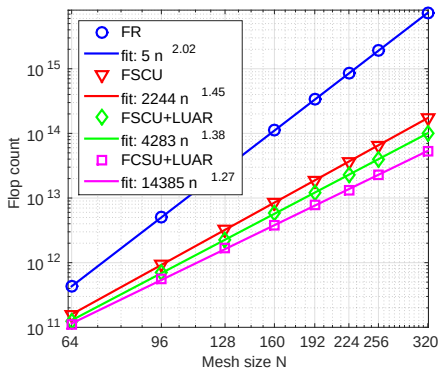
Nested Dissection ordering (geometric)



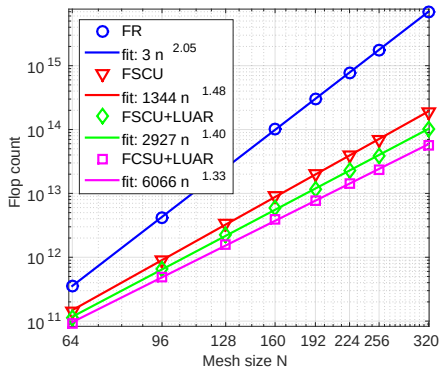
- good agreement with theoretical complexity ($O(n^2)$, $O(n^{1.67})$, $O(n^{1.55})$, and $O(n^{1.33})$)

Experimental MF flop complexity: Poisson ($\varepsilon = 10^{-10}$)

Nested Dissection ordering (geometric)



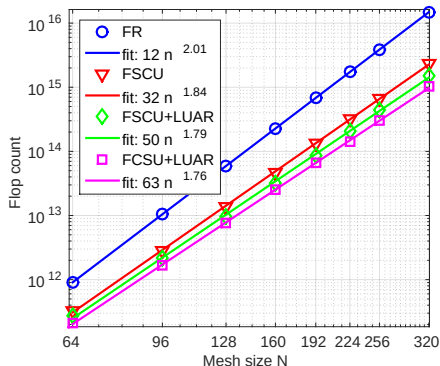
METIS ordering (purely algebraic)



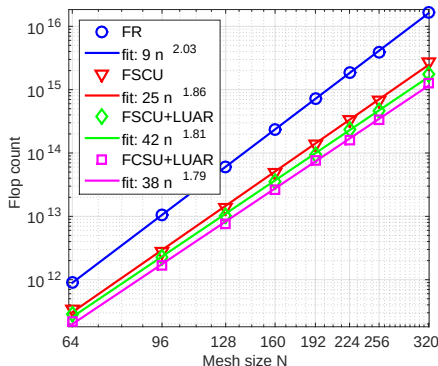
- good agreement with theoretical complexity ($O(n^2)$, $O(n^{1.67})$, $O(n^{1.55})$, and $O(n^{1.33})$)
- remains close to ND complexity with METIS ordering

Experimental MF flop complexity: Helmholtz ($\varepsilon = 10^{-4}$)

Nested Dissection
ordering (geometric)



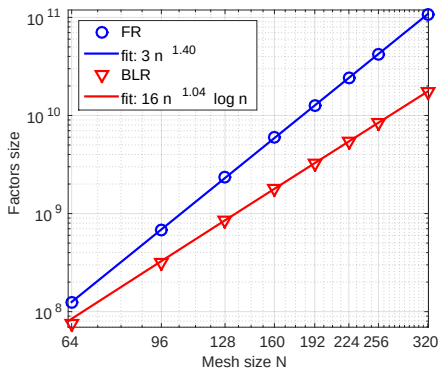
METIS ordering
(purely algebraic)



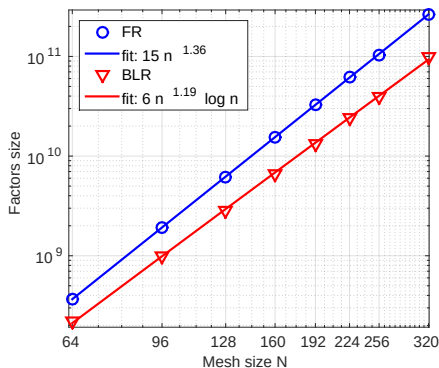
- good agreement with theoretical complexity ($O(n^2)$, $O(n^{1.83})$, $O(n^{1.78})$, and $O(n^{1.67})$)
- remains close to ND complexity with METIS ordering

Experimental MF complexity: factor size

NNZ (Poisson)



NNZ (Helmholtz)



- good agreement with theoretical complexity (FR: $O(n^{1.33})$; BLR: $O(n \log n)$ and $O(n^{1.17} \log n)$)

Experiments are done on the **shared-memory** machines of the LIP laboratory of Lyon:

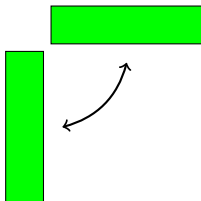
1. **brunch**

- Four Intel(r) 24-cores Broadwell @ **2,2 GHz**
- Peak per core is **35.2 GF/s**
- Total memory is **1.5 TB**

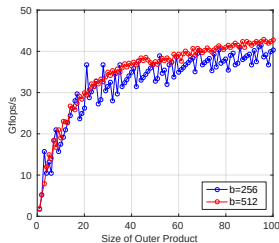
2. **grunch**

- Two Intel(r) 14-cores Haswell @ **2,3 GHz**
- Peak per core is **36.8 GF/s**
- Total memory is **768 GB**

Performance of Outer Product with LUA(R) (24 threads)



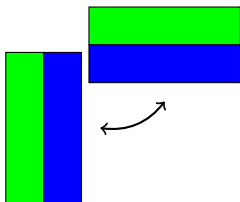
Double complex (z) performance
benchmark of Outer Product



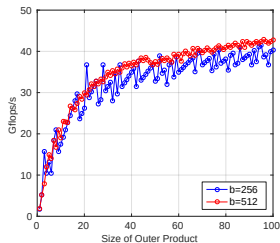
		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead

Performance of Outer Product with LUA(R) (24 threads)



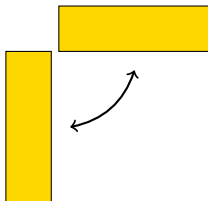
Double complex (z) performance
benchmark of Outer Product



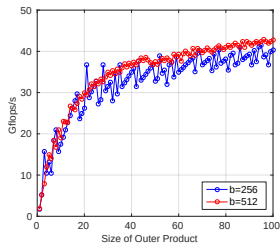
		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead

Performance of Outer Product with LUA(R) (24 threads)



Double complex (z) performance
benchmark of Outer Product



		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead