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
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 $B = XSY$: 

$$B = X \quad S \quad Y$$
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 $B = XSY$:

$$B = X_1 S_1 Y_1 + X_2 S_2 Y_2$$

with $S_1(k, k) = \sigma_k > \varepsilon$, $S_2(1, 1) = \sigma_{k+1} \leq \varepsilon$

If $\tilde{B} = X_1 S_1 Y_1$ then $\|B - \tilde{B}\|_2 = \|X_2 S_2 Y_2\|_2 = \sigma_{k+1} \leq \varepsilon$
Take a dense matrix $B$ of size $b \times b$ and compute its SVD $B = XSY$:

$$B = X_1 S_1 Y_1 + X_2 S_2 Y_2$$

with

$$S_1(k, k) = \sigma_k > \varepsilon, \quad S_2(1, 1) = \sigma_{k+1} \leq \varepsilon$$

If $\tilde{B} = X_1 S_1 Y_1$ then

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

If the singular values of $B$ decay very fast (e.g. exponentially) then $k \ll b$ even for very small $\varepsilon$ (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 $B$.
Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**

A block $B$ represents the interaction between two subdomains $\sigma$ and $\tau$. If they have a **small diameter** and are far away their interaction is weak $\Rightarrow$ rank is low.
Frontal matrices are not low-rank but in some applications they exhibit low-rank blocks.

A block $B$ represents the interaction between two subdomains $\sigma$ and $\tau$. 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 $\sigma$ and $\tau$. If they have a small diameter and are far away their interaction is weak $\Rightarrow$ rank is low.
Frontal matrices are not low-rank but in some applications they exhibit low-rank blocks.

A block $B$ represents the interaction between two subdomains $\sigma$ and $\tau$. If they have a small diameter and are far away their interaction is weak $\Rightarrow$ rank is low.
Frontal matrices are not low-rank but in some applications they exhibit low-rank blocks.

A block $B$ represents the interaction between two subdomains $\sigma$ and $\tau$. 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 $\sigma$ and $\tau$. If they have a **small diameter** and are far away their interaction is weak $\Rightarrow$ rank is low.
Frontal matrices are not low-rank but in some applications they exhibit **low-rank blocks**

A block $B$ represents the interaction between two subdomains $\sigma$ and $\tau$. If they have a **small diameter** and are far away their interaction is weak $\Rightarrow$ rank is low.
$\mathcal{H}$ and BLR matrices

\begin{align*}
\mathcal{H}-\text{matrix} & \\
\text{BLR matrix} & 
\end{align*}
\( \mathcal{H} \) and BLR matrices

\( \mathcal{H} \)-matrix

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

BLR matrix

- Theoretical complexity? \( \Rightarrow O(n^{4/3}) \), as we will prove
- Simple structure

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

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

\[ \mathcal{H} \text{-matrix} \]

- Theoretical complexity? \( \Rightarrow O(n^{4/3}) \), as we will prove
- Simple structure

\[ \text{BLR matrix} \]

Find a good compromise between complexity and performance
\( \mathcal{H} \) and BLR matrices

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

**Find a good comprise between complexity and performance**

⇒ Ongoing collaboration with STRUMPACK team (LBNL) to compare BLR and hierarchical formats

\( \mathcal{H} \)-matrix

BLR matrix

- Theoretical complexity? \( \Rightarrow O(n^{4/3}) \), as we will prove
- Simple structure

Sparse Days, 6-8 Sep. 2017, Toulouse
Applications
3D Seismic Modeling

Helmholtz equation

Single complex (c) arithmetic

Unsymmetric LU factorization

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

Credits: SEISCOPE

<table>
<thead>
<tr>
<th>matrix</th>
<th>n</th>
<th>nnz</th>
<th>flops</th>
<th>storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>5Hz</td>
<td>2.9M</td>
<td>70M</td>
<td>65.0 TF</td>
<td>59.7 GB</td>
</tr>
<tr>
<td>7Hz</td>
<td>7.2M</td>
<td>177M</td>
<td>404.2 TF</td>
<td>205.0 GB</td>
</tr>
<tr>
<td>10Hz</td>
<td>17.2M</td>
<td>446M</td>
<td>2.6 PF</td>
<td>710.8 GB</td>
</tr>
</tbody>
</table>

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)

3D Electromagnetic Modeling
Maxwell equation
Double complex (z) arithmetic
Symmetric $LDL^T$ factorization
Required accuracy: $\varepsilon = 10^{-7}$
Credits: EMGS

<table>
<thead>
<tr>
<th>matrix</th>
<th>n</th>
<th>nnz</th>
<th>flops</th>
<th>storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>E3</td>
<td>2.9M</td>
<td>37M</td>
<td>57.9 TF</td>
<td>77.5 GB</td>
</tr>
<tr>
<td>E4</td>
<td>17M</td>
<td>226M</td>
<td>1.8 PF</td>
<td>1.7 TB</td>
</tr>
<tr>
<td>S3</td>
<td>3.3M</td>
<td>43M</td>
<td>78.0 TF</td>
<td>94.6 GB</td>
</tr>
<tr>
<td>S4</td>
<td>21M</td>
<td>266M</td>
<td>2.5 PF</td>
<td>2.1 TB</td>
</tr>
</tbody>
</table>

Full-Rank statistics

3D Structural Mechanics
Double real (d) arithmetic
Symmetric $LDL^T$ factorization
Required accuracy: $\varepsilon = 10^{-9}$
Credits: Code_Aster (EDF)

<table>
<thead>
<tr>
<th>matrix</th>
<th>n</th>
<th>nnz</th>
<th>flops</th>
<th>storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>perf008d</td>
<td>1.9M</td>
<td>81M</td>
<td>101.0 TF</td>
<td>52.6 GB</td>
</tr>
<tr>
<td>perf008ar</td>
<td>3.9M</td>
<td>159M</td>
<td>377.5 TF</td>
<td>129.8 GB</td>
</tr>
<tr>
<td>perf008cr</td>
<td>7.9M</td>
<td>321M</td>
<td>1.6 PF</td>
<td>341.1 GB</td>
</tr>
<tr>
<td>perf009ar</td>
<td>5.4M</td>
<td>209M</td>
<td>23.4 TF</td>
<td>40.2 GB</td>
</tr>
</tbody>
</table>

Full-Rank statistics
The Block-Low Rank Factorization
Standard BLR factorization: FSCU

- FSCU
Standard BLR factorization: FSCU

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

Sparse Days, 6-8 Sep. 2017, Toulouse
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 ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
  - ⇒ 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 ⇒ **complexity reduction**: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
  - 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 ⇒ complexity reduction: \( O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}}) \)
    ⇒ Collaboration with LSTC to design efficient recompression strategies
FCSU variant: compress before solve

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

- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
    ⇒ 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
FCSU variant: compress before solve

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

**Complexity mainly depends on** $r_{\text{max}}$, **the maximal rank of the blocks**

With $\mathcal{H}$ partitioning, $r_{\text{max}}$ is small

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

**Solution:** extend the theory by bounding the number of full-rank blocks

Until recently, BLR complexity was unknown. Can we use $\mathcal{H}$ theory on BLR matrices?

Complexity mainly depends on $r_{\text{max}}$, the maximal rank of the blocks.

With $\mathcal{H}$ partitioning, $r_{\text{max}}$ 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)
**H vs. BLR complexity**

Until recently, BLR complexity was unknown. Can we use H theory on BLR matrices?

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>operations (OPC)</th>
<th>factor size (NNZ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( r = O(1) )</td>
<td>( r = O(N) )</td>
</tr>
<tr>
<td>FR</td>
<td>( O(n^2) )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td>BLR</td>
<td>( O(n^{4/3}) - O(n^{5/3}) ) ( O(n^{5/3}) - O(n^{11/6}) )</td>
<td>( O(n \log n) ) ( O(n^{7/6} \log n) )</td>
</tr>
<tr>
<td>( \mathcal{H} )</td>
<td>( O(n \log n) )</td>
<td>( O(n^{4/3} \log n) )</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Operations (OPC)</th>
<th>Factor size (NNZ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( r = O(1) )</td>
<td>( r = O(N) )</td>
</tr>
<tr>
<td><strong>FR</strong></td>
<td>( O(n^2) )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td><strong>BLR</strong></td>
<td>( O\left( n^{\frac{4}{3}} \right) - O\left( n^{\frac{5}{3}} \right) )</td>
<td>( O\left( n^{\frac{5}{3}} \right) - O\left( n^{\frac{11}{6}} \right) )</td>
</tr>
<tr>
<td><strong>( \mathcal{H} )</strong></td>
<td>( O(n \log n) )</td>
<td>( O\left( n^{\frac{4}{3}} \log n \right) )</td>
</tr>
</tbody>
</table>

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
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
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 ⇒ 1.9 ratio
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 ⇒ 1.9 ratio

- **Left-looking factorization** reduces the volume of memory transfer in BLR (“communication-avoiding”) ⇒ 2.4 ratio
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 ⇒ 1.9 ratio
- Left-looking factorization reduces the volume of memory transfer in BLR (“communication-avoiding”) ⇒ 2.4 ratio
- Accumulation (LUA) ⇒ 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 ⇒ 1.9 ratio

- Left-looking factorization reduces the volume of memory transfer in BLR (“communication-avoiding”) ⇒ 2.4 ratio

- Accumulation (LUA) ⇒ 2.5 ratio

- Recompression (LUAR) ⇒ 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 ⇒ **1.9 ratio**

- **Left-looking factorization** reduces the volume of memory transfer in BLR (“communication-avoiding”) ⇒ **2.4 ratio**

- **Accumulation** (LUA) ⇒ **2.5 ratio**

- **Recompression** (LUAR) ⇒ **2.6 ratio**

- **Compress before Solve** (FCSU) ⇒ **3.6 ratio**
Multicore performance results (24 threads)

![Normalized time (FR=1)](image)

FCSU variant: compress before solve

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

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

$$\| \tilde{B}_{:,k}\|_{\max} \leq \| \tilde{B}_{:,k}\|_2 = \| XY^T_{:,k}\|_2 = \| Y^T_{:,k}\|_2,$$

assuming $X$ is orthonormal (e.g. RRQR, SVD).
Distributed-memory BLR factorization
Strong scalability analysis

- Flops reduced by 12.8 but volume of communications only by 2.2 ⇒ 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

FR case: LU messages dominate
BLR case: CB messages dominate

CB compression allows for truly reducing the communication. Represents an overhead cost but may lead to speedups depending on network speed w.r.t. processor speed.
Volume of LU messages is reduced in BLR (compressed factors)

Volume of CB messages can be reduced by compressing the CB but it is an overhead cost
• FR case: $LU$ messages dominate
FR case: *LU* messages dominate

BLR case: *CB* messages dominate ⇒ underwhelming reduction of comms.
Communication analysis

- 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
## Software

- **MUMPS 5.1.0**

## Publications


## 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, $\omega$ 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 $s(x, \omega)$.

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

$\omega$ is fixed and equal to 4Hz.
Experimental MF flop complexity: Poisson ($\varepsilon = 10^{-10}$)

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

- 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

<table>
<thead>
<tr>
<th></th>
<th>LL</th>
<th>LUA</th>
<th>LUAR*</th>
</tr>
</thead>
<tbody>
<tr>
<td>average size of Outer Product</td>
<td>16.5</td>
<td>61.0</td>
<td>32.8</td>
</tr>
<tr>
<td>flops ($\times 10^{12}$)</td>
<td>Outer Product Total: 3.76 3.76 1.59</td>
<td>10.19 10.19 8.15</td>
<td></td>
</tr>
<tr>
<td>time (s)</td>
<td>Outer Product Total: 21 14 6</td>
<td>175 167 160</td>
<td></td>
</tr>
</tbody>
</table>

* All metrics include the Recompression overhead

Sparse Days, 6-8 Sep. 2017, Toulouse
Performance of Outer Product with LUA(R) (24 threads)

Double complex (z) performance benchmark of Outer Product

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

* All metrics include the Recompression overhead
Performance of Outer Product with LUA(R) (24 threads)

Double complex (z) performance benchmark of Outer Product

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

* All metrics include the Recompression overhead

Sparse Days, 6-8 Sep. 2017, Toulouse