Sparse direct solvers towards seismic imaging of large 3D domains

P. Amestoy*,1  A. Buttari*,2  J.-Y. L’Excellent†,3  T. Mary*,1

*Université de Toulouse  †ENS Lyon  1IRIT  2CNRS-IRIT  3Inria-LIP

78th EAGE Conference, Vienna 2016
Outline

Sparse direct solver - introduction

Block Low-rank to reduce complexity of direct methods?

Complexity of Block Low-Rank factorization

Performance analysis

Exploiting large sparse RHS

Concluding remarks
Sparse direct solvers

\[ \mathbf{A} \mathbf{x} = \mathbf{b}, \mathbf{A} \text{ large and sparse}, \mathbf{b} \text{ dense or sparse} \]

Sparse direct methods: \( \mathbf{A} = \mathbf{L} \mathbf{U} (\mathbf{L} \mathbf{D} \mathbf{L}^T) \)

on multiprocessor architectures

Frequency domain FWI
Helmholtz equations
Complex large sparse matrix \( \mathbf{A} \)
Multiple (very) sparse \( \mathbf{B} \)

(3D EAGE/SEG overthrust model)
Sparse direct solvers

Discretization of a physical problem
(e.g. Code_Aster, finite elements)

Solution of sparse systems
\[ A \, X = B \]

Often a significant part of simulation cost

Main steps:
- Preprocess \( A \) and \( B \)
- Factor \( A = LU \) (\( LDL^T \) if \( A \) symmetric)
- Triangular solve: \( LY = B \), then \( UX = Y \)

Preferred to iterative methods for their robustness, accuracy, and capacity to solve efficiently multiple/successive right-hand sides
Sparse direct solvers: black boxes?

Matrix properties and preprocessing influence:
- Size of $L, U$ and memory
- Operation count and time
- Numerical accuracy

Original ($A = \text{LHR01}$)  Preprocessed matrix ($A'(\text{LHR01})$)

Modified problem: $A'x' = b'$ with $A' = PD_rAQQD_cP^T$
Multifrontal method [Duff Reid ’83]

Memory is divided into two parts:

- the factors
- the active memory

Elimination tree represents dependencies between tasks
• **Assume:**
  - $n = N^3$ degrees of freedom,
  - $N^2$ seismic sources
  - $N$ time steps

• **Time domain FWI** scales to $O(N^6)$ (Plessix, 2007)

• **Frequency domain FWI**...
  - Factorization of one matrix (one frequency) scales to $O(N^6)$
  - Size of $LU$ factors scales to $O(N^4)$ and $N^2$ sources/RHS
    $\Rightarrow$ Solution scales to $O(N^6)$

...if only few discrete frequencies required (case of wide-azimuth long-offset (OBC/OBN) surveys) then frequency domain FWI scales to $O(N^6)$
Questions addressed in this talk

- How to reduce the complexity of direct methods? (i.e., in $\mathcal{O}(N^\alpha)$, with $\alpha < 6$)

- How to translate complexity reduction into a performance gain in a parallel setting (shared and/or distributed)?

- How to efficiently process multiple sparse right-hand sides?
Outline

Sparse direct solver - introduction

Block Low-rank to reduce complexity of direct methods?

Complexity of Block Low-Rank factorization

Performance analysis

Exploiting large sparse RHS

Concluding remarks
Application specific solvers: BLR feature

- Applicative context: discretized PDEs, integral equations
- BLR factorization computes an approximation $A = L_\varepsilon U_\varepsilon$ at accuracy $\varepsilon$ controlled by the user
- Operations and factor size reduction

Work supported by PhD thesis: C. Weisbecker (2010-2013, supported by EDF) and T. Mary (2014-ongoing)

Main features of Block Low Rank (BLR) format

- Algebraic robust solver; flat and simple format
- Compatibility with numerical pivoting
- Variants of BLR can reach complexity as low as non-fully structured $\mathcal{H}$ format

⇒ Many representations: Recursive $\mathcal{H}, \mathcal{H}^2$ [Bebendorf, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...], BLR ...
A block $B$ represents the interaction between two subdomains. If they have a small diameter and are far away, their interaction is weak $\Rightarrow$ rank is low.

$$\tilde{B} = XY^T$$ such that $\text{rank}(\tilde{B}) = k_\varepsilon$ and $\|B - \tilde{B}\| \leq \varepsilon$

If $k_\varepsilon \ll \text{size}(B)$ $\Rightarrow$ memory and flops can be reduced with a controlled loss of accuracy ($\leq \varepsilon$)
Block Low Rank multifrontal solver

Elimination tree

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

Elimination tree

\[ B = X_1 S_1 Y_1 + X_2 S_2 Y_2 \]

\[ \| E \|_2 = \| X_2 S_2 Y_2 \|_2 = \sigma_{k+1} \leq \varepsilon \]
Application to frequency-domain seismic modeling

from left to right: FR, $\varepsilon = 10^{-5}$, $\varepsilon = 10^{-4}$, $\varepsilon = 10^{-3}$ (overthrust model)

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>fqcy</th>
<th>ops</th>
<th>memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(10^{-5})$</td>
<td>2 Hz</td>
<td>41.8 %</td>
<td>61.8 %</td>
</tr>
<tr>
<td></td>
<td>4 Hz</td>
<td>27.4 %</td>
<td>50.0 %</td>
</tr>
<tr>
<td></td>
<td>8 Hz</td>
<td>21.8 %</td>
<td>41.6 %</td>
</tr>
<tr>
<td>$(10^{-4})$</td>
<td>2 Hz</td>
<td>32.9 %</td>
<td>53.4 %</td>
</tr>
<tr>
<td></td>
<td>4 Hz</td>
<td>20.0 %</td>
<td>42.2 %</td>
</tr>
<tr>
<td></td>
<td>8 Hz</td>
<td>15.2 %</td>
<td>28.9 %</td>
</tr>
</tbody>
</table>

% : percentage of standard (full-rank) sparse solver, [SEG’13 proceedings]
Sparse direct solver - introduction

Block Low-rank to reduce complexity of direct methods?

Complexity of Block Low-Rank factorization

Performance analysis

Exploiting large sparse RHS

Concluding remarks
Context of the study:

- **Extended theoretical work** on $\mathcal{H}$-matrices by Hackbush and Bebendorf (2003) and Bebendorf (2005, 2007) to the BLR case
  

- Discretized elliptic PDEs on a cubic domain of size $N$ (i.e., $n = N^3$)

- **Two BLR variants:**
  - BLR: original version (Phd of C. Weisbecker (2013))
  - BLR+: new variants, more efficient and with lower complexity

- **Two families of equations:**
  - $r = \mathcal{O}(1)$: rank of off-diagonal blocks bound by a constant.
    - Example: the Poisson equation
  - $r = \mathcal{O}(N)$: rank of off-diagonal blocks bound by $N$.
    - Example: the Helmholtz equation
## 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>$r = \mathcal{O}(1)$</td>
<td>$r = \mathcal{O}(N)$</td>
</tr>
<tr>
<td><strong>FR</strong></td>
<td>$\mathcal{O}(N^6)$</td>
<td>$\mathcal{O}(N^6)$</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(N^4)$</td>
<td>$\mathcal{O}(N^4)$</td>
</tr>
<tr>
<td><strong>BLR</strong></td>
<td>$\mathcal{O}(N^5)$</td>
<td>$\mathcal{O}(N^{5.5})$</td>
</tr>
<tr>
<td><strong>BLR+</strong></td>
<td>$\mathcal{O}(N^4)$</td>
<td>$\mathcal{O}(N^5)$</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(N^3 \log N)$</td>
<td>$\mathcal{O}(N^{3.5} \log N)$</td>
</tr>
<tr>
<td><strong>H</strong></td>
<td>$\mathcal{O}(N^4)$</td>
<td>$\mathcal{O}(N^5)$</td>
</tr>
<tr>
<td><strong>H</strong> (fully struct.)</td>
<td>$\mathcal{O}(N^3)$</td>
<td>$\mathcal{O}(N^4)$</td>
</tr>
</tbody>
</table>

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

**Important properties**: with both $r = \mathcal{O}(1)$ or $r = \mathcal{O}(N)$

- Complexity of the original BLR has a **lower exponent** than the full-rank
- Variants improves complexity, (BLR+) being not so far from the $\mathcal{H}$-case
Experimental MF flop complexity: Helmholtz ($\varepsilon = 10^{-4}$)

- Good agreement with theoretical complexity ($O(N^6)$, $O(N^{5.5})$, and $O(N^5)$)
- Purely algebraic approach (METIS) achieves comparable complexity to geometric (ND)
Sparse direct solver - introduction

Block Low-rank to reduce complexity of direct methods?

Complexity of Block Low-Rank factorization

Performance analysis

Exploiting large sparse RHS

Concluding remarks
1. **MUMPS sparse solver** used for all the experiments (http://mumps-solver.org/)

2. **Distributed memory** experiments are done on the eos supercomputer at the CALMIP center of Toulouse (grant 2014-P0989):
   - Two Intel(r) 10-cores Ivy Bridge @ 2.8 GHz
   - Peak per core is 22.4 GF/s (real, double precision)
   - 64 GB memory per node
   - Infiniband FDR interconnect

3. **Shared memory** experiments are done on grunch at the LIP laboratory of Lyon:
   - Two Intel(r) 14-cores Haswell @ 2.3 GHz
   - Peak per core is 36.8 GF/s (real, double precision)
   - Total memory is 768 GB
Performance on seismic modeling on 640 cores

3D seismic Modeling
North Sea case study
(Simple) Complex matrix
Helmholtz equation
SEISCOPE project

Matrix from 3D FWI for seismic modeling (credits: SEISCOPE)

<table>
<thead>
<tr>
<th>matrix</th>
<th>n</th>
<th>nnz</th>
<th>MUMPS (Full-Rank)</th>
<th>BLR* time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10Hz/35m</td>
<td>17M</td>
<td>446M</td>
<td>1132s</td>
<td>324s</td>
</tr>
</tbody>
</table>

*ε = 10^{-3}; **estimated speedup on 64 × 10 cores
3D Electromagnetic Modeling
(Double) Complex matrix

Matrix D4 requires:
3 TBytes of storage, 3 PetaFlops

Matrix from 3D EM problems (credits: EMGS)

<table>
<thead>
<tr>
<th>matrix</th>
<th>n</th>
<th>nnz</th>
<th>MUMPS-(Full-Rank)</th>
<th>BLR*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>D4</td>
<td>30M</td>
<td>384M</td>
<td>2221s</td>
<td>566s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>373</td>
<td>33%</td>
</tr>
</tbody>
</table>

\( \varepsilon = 10^{-7}; \) **estimated speedup on 90 \times 10 cores
Gains due to BLR (distributed, MPI+OpenMP)

Poisson ($\varepsilon = 10^{-6}$)

- gains increase with problem size
- gain in flops does not fully translate into gain in time
- multithreaded efficiency lower with BLR than with Full-Rank (FR)
- same remarks apply to Helmholtz, to a lesser extent
Gains due to BLR (distributed, MPI+OpenMP)

- gains increase with problem size
- gain in flops does not fully translate into gain in time
- multithreaded efficiency lower with BLR than with Full-Rank (FR)
- same remarks apply to Helmholtz, to a lesser extent

⇒ improve multithreading behaviour
Performance analysis (shared memory, 28 threads)

\[ \%_{ci} \]

\[ \%_{nci} \]

<table>
<thead>
<tr>
<th>1 thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
</tr>
<tr>
<td>FR</td>
</tr>
</tbody>
</table>

3D Poisson; \( n = 256^3 \) (16M); \( \epsilon = 10^{-6} \)
Performance analysis (shared memory, 28 threads)

<table>
<thead>
<tr>
<th></th>
<th>1 thread</th>
<th>28 threads</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>%(ci)</td>
</tr>
<tr>
<td>FR</td>
<td>62660s (1)</td>
<td>1%</td>
</tr>
<tr>
<td>BLR</td>
<td>7823s (8)</td>
<td>11%</td>
</tr>
</tbody>
</table>

3D Poisson; \(n = 256^3\) (16M); \(\varepsilon = 10^{-6}\)

PhD W. Sid Lakhdar (2014)

78th EAGE Conference, Vienna 2016
Performance analysis (shared memory, 28 threads)

Computationally Intensive

Not Computationally Intensive

<table>
<thead>
<tr>
<th></th>
<th>1 thread</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>%(n_{ci})</td>
</tr>
<tr>
<td>FR</td>
<td>62660s (1)</td>
<td>1%</td>
</tr>
<tr>
<td>BLR</td>
<td>7823s (8)</td>
<td>11%</td>
</tr>
<tr>
<td>BLR+</td>
<td>2464s (25)</td>
<td>38%</td>
</tr>
</tbody>
</table>

3D Poisson; \(n = 256^3\) (16M); \(\varepsilon = 10^{-6}\)
### Performance analysis (shared memory, 28 threads)

![Performance Analysis Diagram]

<table>
<thead>
<tr>
<th></th>
<th>1 thread</th>
<th>28 threads</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>%(n_{ci})</td>
</tr>
<tr>
<td>FR</td>
<td>62660s (1)</td>
<td>1%</td>
</tr>
<tr>
<td>BLR</td>
<td>7823s (8)</td>
<td>11%</td>
</tr>
<tr>
<td>BLR+</td>
<td>2464s (25)</td>
<td>38%</td>
</tr>
</tbody>
</table>

3D Poisson; \(n = 256^3\) (16M); \(\varepsilon = 10^{-6}\);
Performance analysis (shared memory, 28 threads)

<table>
<thead>
<tr>
<th></th>
<th>1 thread</th>
<th>28 threads</th>
<th>28 threads + L0 OMP*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>%(\text{n_ci})</td>
<td>time</td>
</tr>
<tr>
<td>FR</td>
<td>62660s (1)</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>BLR</td>
<td>7823s (8)</td>
<td>11%</td>
<td></td>
</tr>
<tr>
<td>BLR+</td>
<td>2464s (25)</td>
<td>38%</td>
<td>557s (7)</td>
</tr>
</tbody>
</table>

3D Poisson; \(n = 256^3\) (16M); \(\varepsilon = 10^{-6}\); *PhD W. Sid Lakhdar (2014)
Performance analysis (shared memory, 28 threads)

<table>
<thead>
<tr>
<th></th>
<th>1 thread</th>
<th>28 threads</th>
<th>28 threads + L0 OMP*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>%(n_{ci})</td>
<td>time</td>
</tr>
<tr>
<td>FR</td>
<td>62660s (1)</td>
<td>1%</td>
<td>3805s (1)</td>
</tr>
<tr>
<td>BLR</td>
<td>7823s (8)</td>
<td>11%</td>
<td>1356s (3)</td>
</tr>
<tr>
<td>BLR+</td>
<td>2464s (25)</td>
<td>38%</td>
<td>557s (7)</td>
</tr>
</tbody>
</table>

3D Poisson; \(n = 256^3\) (16M); \(\varepsilon = 10^{-6}\) ; *PhD W. Sid Lakhdar (2014)
Improved performance relies on new BLR variants and improved multithreading based on Sid-Lakhdar’s PhD (2011-2014) so called L0 OMP thread

<table>
<thead>
<tr>
<th>application</th>
<th>matrix</th>
<th>L0 OMP&lt;sup&gt;a&lt;/sup&gt;</th>
<th>time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electro-magnetism&lt;sup&gt;†&lt;/sup&gt;</td>
<td>E3</td>
<td>no</td>
<td>451</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>393</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>no</td>
<td>585</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>519</td>
</tr>
<tr>
<td>Structural mechanics&lt;sup&gt;‡&lt;/sup&gt;</td>
<td>perf008d</td>
<td>no</td>
<td>249</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>208</td>
</tr>
<tr>
<td></td>
<td>perf008ar</td>
<td>no</td>
<td>831</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>787</td>
</tr>
</tbody>
</table>

*estimated (ongoing work)

<sup>†</sup> Credits: EMGS (ε = 10<sup>−7</sup>)

<sup>‡</sup> Credits: Code_Aster (ε = 10<sup>−9</sup>)

<sup>a</sup> W. Sid-Lakhdar's PhD (2011-2014)

<sup>b</sup> C. Weisbecker's PhD (2010-2013)

<sup>c</sup> T. Mary's PhD (2014-ongoing)
Outline

Sparse direct solver - introduction

Block Low-rank to reduce complexity of direct methods?

Complexity of Block Low-Rank factorization

Performance analysis

Exploiting large sparse RHS

Concluding remarks
Exploiting sparsity of right-hand sides

Context

- \( \mathbf{LU} \mathbf{x} = \mathbf{b}, \mathbf{Ly} = \mathbf{b}, \mathbf{Ux} = \mathbf{y} \)
- Sparse \( \mathbf{y} \rightarrow \) not all of the tree/factors need be used [Gilbert, 1994] (similar property for partial solution)
- Typically found in electromagnetism, geophysics, explicit Schur, refactoring ...

(a) Solve \( \mathbf{Ly} = \mathbf{b} \)

(b) Elimin. tree
Tree pruning to minimize flops

- Group columns "close in the tree" to limit flops

Questions:
- Columns "close in the tree"?
- How to expose parallelism?
Need for grouping / permuting columns:
  - "Close in the tree"? dependent on the application and on the tree structure
  - Combinatorial problem $\rightarrow$ similarity with computing entries in $A^{-1}$

On going work, Phd thesis of Gilles Moreau (ENS-Lyon) with applications from seismic modeling and electromagnetism
Outline

- Sparse direct solver - introduction
- Block Low-rank to reduce complexity of direct methods?
- Complexity of Block Low-Rank factorization
- Performance analysis
- Exploiting large sparse RHS
- Concluding remarks
Concluding remarks

3D Frequency domain Full-Wave Inversion

- **Theoretical gains:** (not yet fully exploited)
  - Factorization $O(N^6) \Rightarrow O(N^5)$
  - Solution Phase ($N^2$ sources/RHS) $O(N^6) \Rightarrow O(N^{5.5}\log N)$

- **North Sea case study** (680 cores):
  - BLR ($\varepsilon = 10^{-4}$) accelerates factorization by a factor of 3

  **Full FWI:** $49 \text{hr} \Rightarrow 36 \text{hr}$ (MUMPS-SEISCOPE research work submitted to Geophysics) [2015]

Perspectives for further improvement:
- Complexity: BLR+ and BLR solution phase
- Exploit sparsity of multiple RHS
- Improve efficiency (MPI and multithreading)
Questions?