# 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 [1]IRIT [2]CNRS-IRIT [3]Inria-LIP

78[th] 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

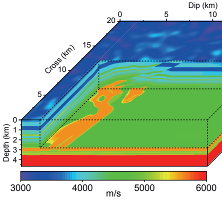
Exploiting large sparse RHS

Concluding remarks

**$\mathbf{A}$ $\mathbf{X}$ = $\mathbf{B}$, $\mathbf{A}$ large and sparse, $\mathbf{B}$ dense or sparse**

Sparse direct methods : $\mathbf{A} = \mathbf{LU}$ ($\mathbf{LDL^T}$)

on multiprocessor architectures




(3D EAGE/SEG overthrust model)

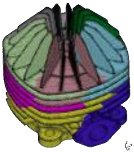
$\Longrightarrow$

Frequency domain FWI

Helmholtz equations

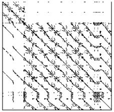
Complex large sparse matrix $\mathbf{A}$

Multiple (very) sparse $\mathbf{B}$



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

$\Longrightarrow$ **Solution of sparse systems**
**A X = B**



*Often a significant part of simulation cost*

Main steps:

- Preprocess **A** and **B**
- Factor $\mathbf{A} = \mathbf{LU}$ ($\mathbf{LDL^T}$ if **A** symmetric)
- Triangular solve: $\mathbf{LY} = \mathbf{B}$, then $\mathbf{UX} = \mathbf{Y}$

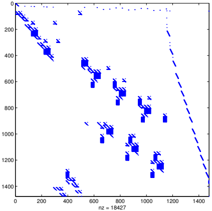
Preferred to iterative methods for their robustness, accuracy, and capacity to solve efficiently multiple/successive right-hand sides

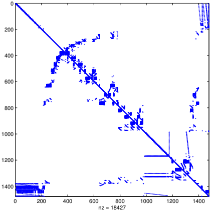
# Sparse direct solvers: black boxes?

Matrix properties and preprocessing influence:

- Size of $L$,$U$ and memory
- Operation count and time
- Numerical accuracy

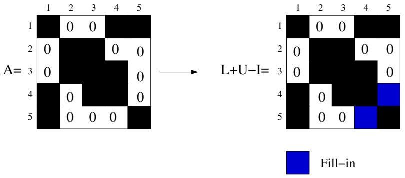
Original ($A =$ LHR01) Preprocessed matrix ($A'$(LHR01))



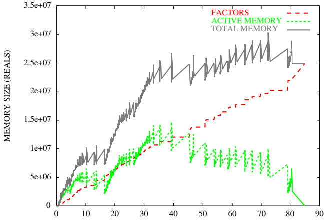

Modified problem: $A'x' = b'$ with $A' = PD_rAQD_cP^T$

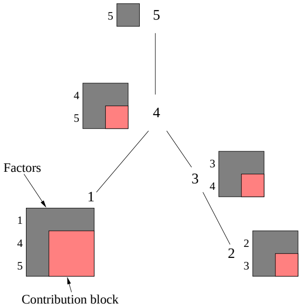
# Multifrontal method [Duff Reid '83]




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 $\mathcal{O}(N^6)$ (Plessix, 2007)
- **Frequency domain FWI**...
  - Factorization of one matrix (one frequency) scales to $\mathcal{O}(N^6)$
  - Size of *LU* factors scales to $\mathcal{O}(N^4)$ and $N^2$ sources/RHS
    $\Longrightarrow$ Solution scales to $\mathcal{O}(N^6)$

...**if only few discrete frequencies required** (case of wide-azimuth long-offset (OBC/OBN) surveys) then frequency domain FWI scales to $\mathcal{O}(N^6)$

# 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 $\mathbf{A} = \mathbf{L}_\varepsilon \mathbf{U}_\varepsilon$ at accuracy $\varepsilon$ controlled by the user
- Operations and factor size reduction

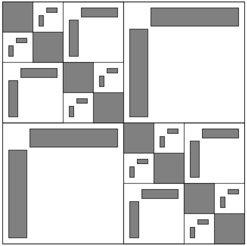
  Work supported by PhD thesis: C. Weisbecker (2010-2013, supported by EDF) and T. Mary (2014-ongoing)

## Main features of Block Low Rank (BLR ) format

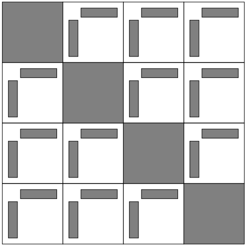
- Algebraic robust solver; flat and simple format
- Compatibility with numerical pivoting
- Variants of BLR can reach complexity as low as non-fully structured $\mathcal{H}$ format

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

# $\mathcal{H}$ and BLR matrices



$\mathcal{H}$-matrix



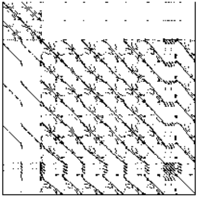
BLR matrix

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.

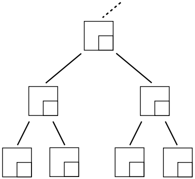
$$\tilde{B} = XY^T \text{ such that } \text{rank}(\tilde{B}) = k_\varepsilon \text{ and } \|B - \tilde{B}\| \leq \varepsilon$$

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

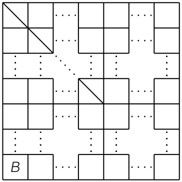
# Block Low Rank multifrontal solver



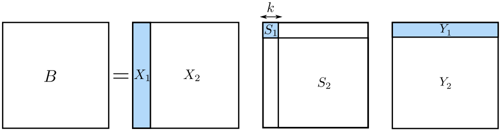
⇒



Elimination tree

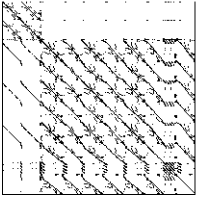



⇒

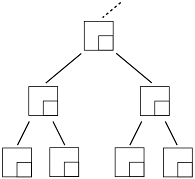


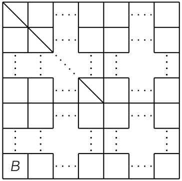

Singular value decomposition (SVD) of each block $B \Rightarrow B = X_1S_1Y_1 + X_2S_2Y_2$

# Block Low Rank multifrontal solver

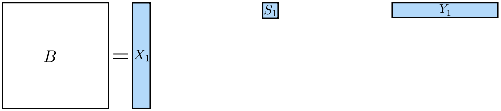


⇒



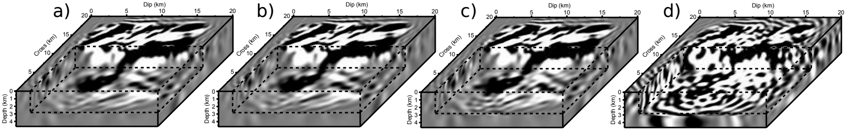

⇒




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

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

# Application to frequency-domain seismic modeling




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

| | | ops | memory | |
|---|---|---|---|---|
| $\varepsilon$ | fqcy | | factors | active mem. |
| $(10^{-5})$ | 2 Hz | 41.8 % | 61.8 % | 32.3% |
| | 4 Hz | 27.4 % | 50.0 % | 24.4% |
| | 8 Hz | 21.8 % | 41.6 % | 23.9% |
| $(10^{-4})$ | 2 Hz | 32.9 % | 53.4 % | 23.9% |
| | 4 Hz | 20.0 % | 42.2 % | 21.7% |
| | 8 Hz | 15.2 % | 28.9 % | 19.4% |

% : percentage of standard (full-rank) sparse solver, [SEG'13 proceedings]

# 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

# Complexity of multifrontal BLR factorization

Context of the study:

- Extended theoretical work on $\mathcal{H}$-matrices by Hackbush and Bebendorf (2003) and Bebendorf (2005, 2007) to the BLR case
  - Amestoy, Buttari, L'Excellent and Mary. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, sumitted to SIAM SISC, 2016.
- 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

| | operations (OPC) | | factor size (NNZ) | |
|---|---|---|---|---|
| | $r = \mathcal{O}(1)$ | $r = \mathcal{O}(N)$ | $r = \mathcal{O}(1)$ | $r = \mathcal{O}(N)$ |
| FR | $\mathcal{O}(N^6)$ | $\mathcal{O}(N^6)$ | $\mathcal{O}(N^4)$ | $\mathcal{O}(N^4)$ |
| BLR | $\mathcal{O}(N^5)$ | $\mathcal{O}(N^{5.5})$ | $\mathcal{O}(N^3 \log N)$ | $\mathcal{O}(N^{3.5} \log N)$ |
| BLR+ | $\mathcal{O}(N^4)$ | $\mathcal{O}(N^5)$ | $\mathcal{O}(N^3 \log N)$ | $\mathcal{O}(N^{3.5} \log N)$ |
| $\mathcal{H}$ | $\mathcal{O}(N^4)$ | $\mathcal{O}(N^5)$ | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^{3.5})$ |
| $\mathcal{H}$ (fully struct.) | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^4)$ | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^{3.5})$ |

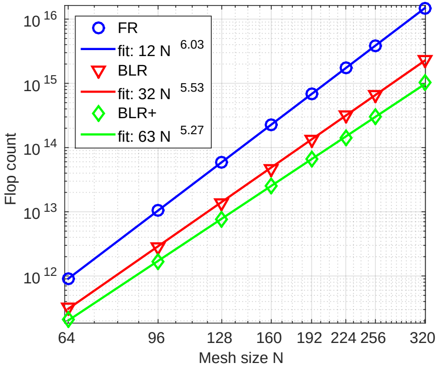
in the 3D case (similar analysis possible for 2D)

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

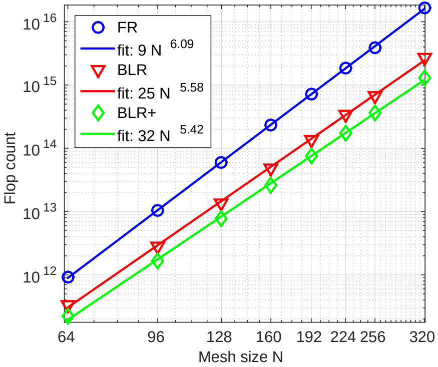
- Complexity of the orginal BLR has a lower exponent than the full-rank
- Variants improves complexity, (BLR+) being not so far from the $\mathcal{H}$-case

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

Nested Dissection ordering (geometric)




METIS ordering (purely algebraic)




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

# 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

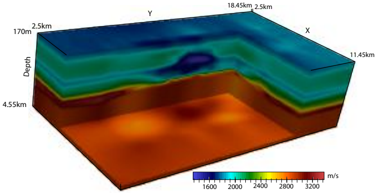
# Experimental Setting

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

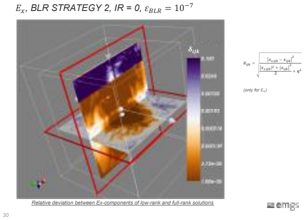
Matrix from 3D FWI for seismic modeling (credits: SEISCOPE)

| matrix | n | nnz | MUMPS (Full-Rank) time | MUMPS (Full-Rank) sp-up** | MUMPS (Full-Rank) $\%_{peak}$ | BLR* time |
|---|---|---|---|---|---|---|
| 10Hz/35m | 17M | 446M | 1132s | 295 | 35% | 324s |

$^*\varepsilon = 10^{-3}$; $^{**}$estimated speedup on $64 \times 10$ cores

# Performance on 3D EM application on 900 cores




3D Electromagnetic Modeling

(Double) Complex matrix

Matrix D4 requires:

3 TBytes of storage, 3 PetaFlops

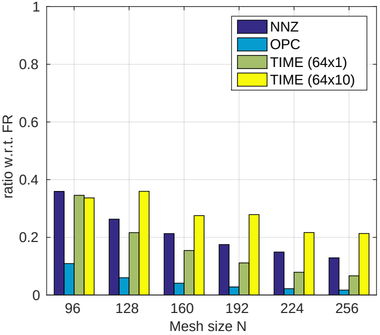
Matrix from 3D EM problems (credits: EMGS)

| matrix | n | nnz | MUMPS-(Full-Rank) time | sp-up** | $\%_{peak}$ | BLR* time |
|---|---|---|---|---|---|---|
| D4 | 30M | 384M | 2221s | 373 | 33% | 566s |

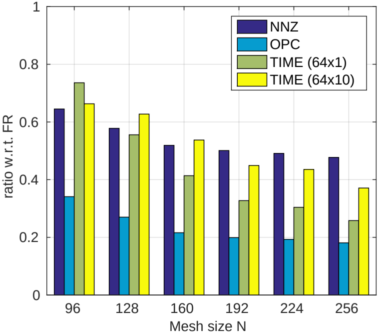
$^*\varepsilon = 10^{-7}$; $^{**}$estimated speedup on $90 \times 10$ cores

# Gains due to BLR (distributed, MPI+OpenMP)

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



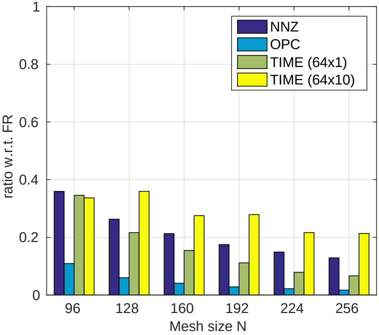

Helmholtz ($\varepsilon = 10^{-4}$)



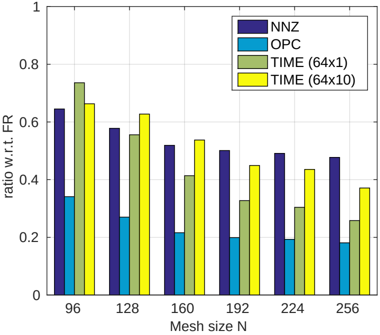

- gains increase with problem size
- gain in flops does not fully translate into gain in time
- multithreaded efficiency lower with BLR than with Full-Rank (FR)
- same remarks apply to Helmoltz, to a lesser extent

# Gains due to BLR (distributed, MPI+OpenMP)

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



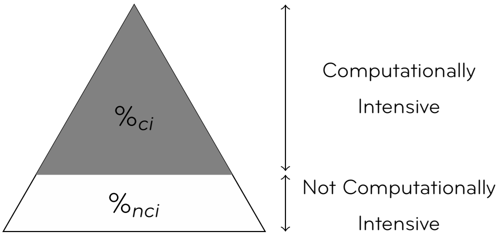

Helmholtz ($\varepsilon = 10^{-4}$)




- 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 Helmoltz, to a lesser extent

$\Rightarrow$ *improve multithreading behaviour*

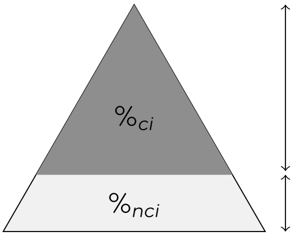
# Performance analysis (shared memory, 28 threads)




| | 1 thread | |
|---|---|---|
| | time | $\%_{nci}$ |
| FR | 62660s ( 1) | 1% |

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

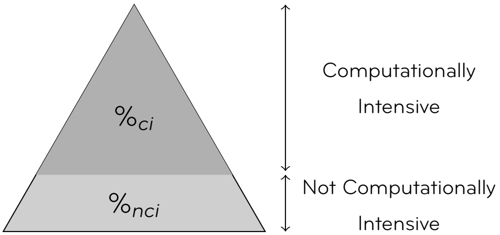
# Performance analysis (shared memory, 28 threads)




| | 1 thread | | | |
|---|---|---|---|---|
| | time | $\%_{nci}$ | | |
| FR | 62660s ( 1) | 1% | | |
| BLR | 7823s ( 8) | 11% | | |

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

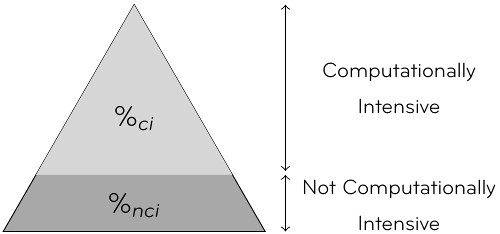
# Performance analysis (shared memory, 28 threads)




| | 1 thread | | | |
|---|---|---|---|---|
| | time | $\%_{nci}$ | | |
| FR | 62660s ( 1) | 1% | | |
| BLR | 7823s ( 8) | 11% | | |
| BLR+ | 2464s (25) | 38% | | |

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

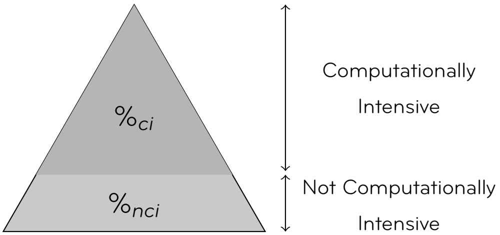
# Performance analysis (shared memory, 28 threads)




| | 1 thread | | 28 threads | |
|---|---|---|---|---|
| | time | $\%_{nci}$ | time | $\%_{nci}$ |
| FR | 62660s ( 1) | 1% | | |
| BLR | 7823s ( 8) | 11% | | |
| BLR+ | 2464s (25) | 38% | 557s (7) | 68% |

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

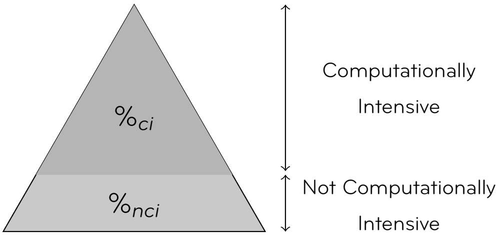
# Performance analysis (shared memory, 28 threads)




| | 1 thread | | 28 threads | | 28 threads + L0 OMP* | |
|---|---|---|---|---|---|---|
| | time | $\%_{nci}$ | time | $\%_{nci}$ | time | $\%_{nci}$ |
| FR | 62660s ( 1) | 1% | | | | |
| BLR | 7823s ( 8) | 11% | | | | |
| BLR+ | 2464s (25) | 38% | 557s (7) | 68% | 310s (11) | 42% |

3D Poisson; $n = 256^3$ (16M); $\varepsilon = 10^{-6}$ ; *PhD W. Sid Lakhdar (2014)

# Performance analysis (shared memory, 28 threads)




| | 1 thread | | 28 threads | | 28 threads + L0 OMP* | |
|---|---|---|---|---|---|---|
| | time | $\%_{nci}$ | time | $\%_{nci}$ | time | $\%_{nci}$ |
| FR | 62660s ( 1) | 1% | 3805s (1) | 9% | 3430s ( 1) | 0% |
| BLR | 7823s ( 8) | 11% | 1356s (3) | 26% | 1160s ( 3) | 14% |
| BLR+ | 2464s (25) | 38% | 557s (7) | 68% | 310s (11) | 42% |

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*

| application | matrix | L0 OMP[a] | time in seconds FR | BLR[b] | BLR+[c] |
|---|---|---|---|---|---|
| Electro-magnetism† | E3 | no | 451 | 265 | 184 |
| | | yes | 393 | 199 | 114 |
| | S3 | no | 585 | 324 | 223 |
| | | yes | 519 | 239 | 136 |
| Structural mechanics‡ | perf008d | no | 249 | 177 | 137 |
| | | yes | 208 | 140* | 100* |
| | perf008ar | no | 831 | 574 | 331 |
| | | yes | 787 | 531* | 287* |

*estimated (ongoing work)

† Credits: EMGS ($\varepsilon = 10^{-7}$)
‡ Credits: Code_Aster ($\varepsilon = 10^{-9}$)

[a] W. Sid-Lakhdar's PhD (2011-2014)
[b] C. Weisbecker's PhD (2010-2013)
[c] 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

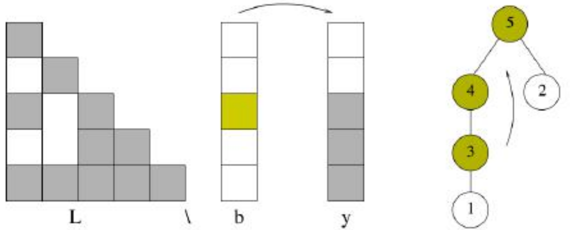
Exploiting large sparse RHS

Concluding remarks

# Exploiting sparsity of right-hand sides

## Context

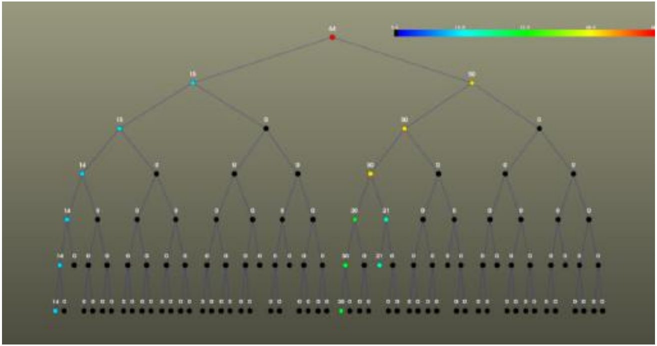
- $\mathbf{LU}x = b, \mathbf{L}y = b, \mathbf{U}x = y$
- Sparse $y \rightarrow$ not all of the tree/factors need be used [Gilbert,1994] (similar property for partial solution)
- Typically found in electromagnetism, geophysics, explicit Schur, refactoring ...




(a) Solve $Ly = b$ (b) Elimin. tree

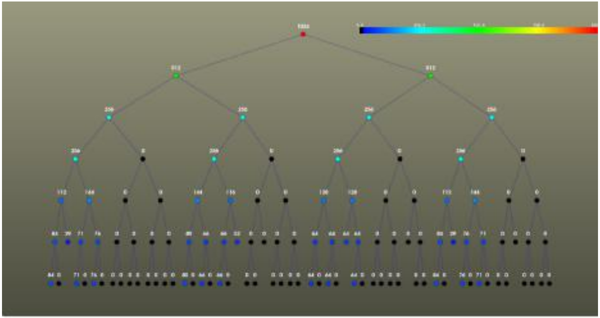
# Tree pruning to minimize flops

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



- Questions:
  - Columns "close in the tree"?
  - How to expose parallelism?

# Exploiting tree parallelism and sparsity of RHS



- Need for grouping / permuting columns:
  - "Close in the tree"? dependent on the application and on the tree structure
  - Combinatorial problem → similarity with computing entries in $\mathbf{A}^{-1}$
- 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

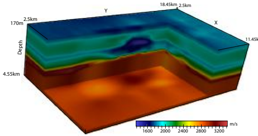
Concluding remarks

# Concluding remarks

## 3D Frequency domain Full-Wave Inversion

- Theoretical gains: (not yet fully exploited)
  - Factorization $\mathcal{O}(N^6) \Rightarrow \mathcal{O}(N^5)$
  - Solution Phase ($N^2$ sources/RHS) $\mathcal{O}(N^6) \Rightarrow \mathcal{O}(N^{5.5} logN)$
- North Sea case study (680 cores):
  - BLR ($\varepsilon = 10^{-4}$) accelerates factorization by a factor of $3$

  Full FWI : $49hr \Rightarrow 36hr$ (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?