emgs

LARGE-SCALE 3D CSEM MODELING WITH A BLR MUMPS SOLVER

Sebastien de la Kethulle de Ryhove, Piyoosh Jaysaval & Daniil V. Shantsev
EMGS R&D, Norway

MUMPS Users Days, Grenoble, France
June 1-2, 2017

Spot the difference.

Our Story

- 2013 MUMPS Users Days (guests)
- 2014-2015 MUMPS – EMGS research project
- 2017 Geophys. J. Intl. article
- 2017 MUMPS Users Days again
- 2014 - … Use of MUMPS in EMGS for research / production

Geophysical Journal International

Geophys. J. Int. (2017) **209,** 1558–1571 doi: 10.1093/gji/ggx106
Advance Access publication 2017 March 15
GJI Geomagnetism, rock magnetism and palaeomagnetism

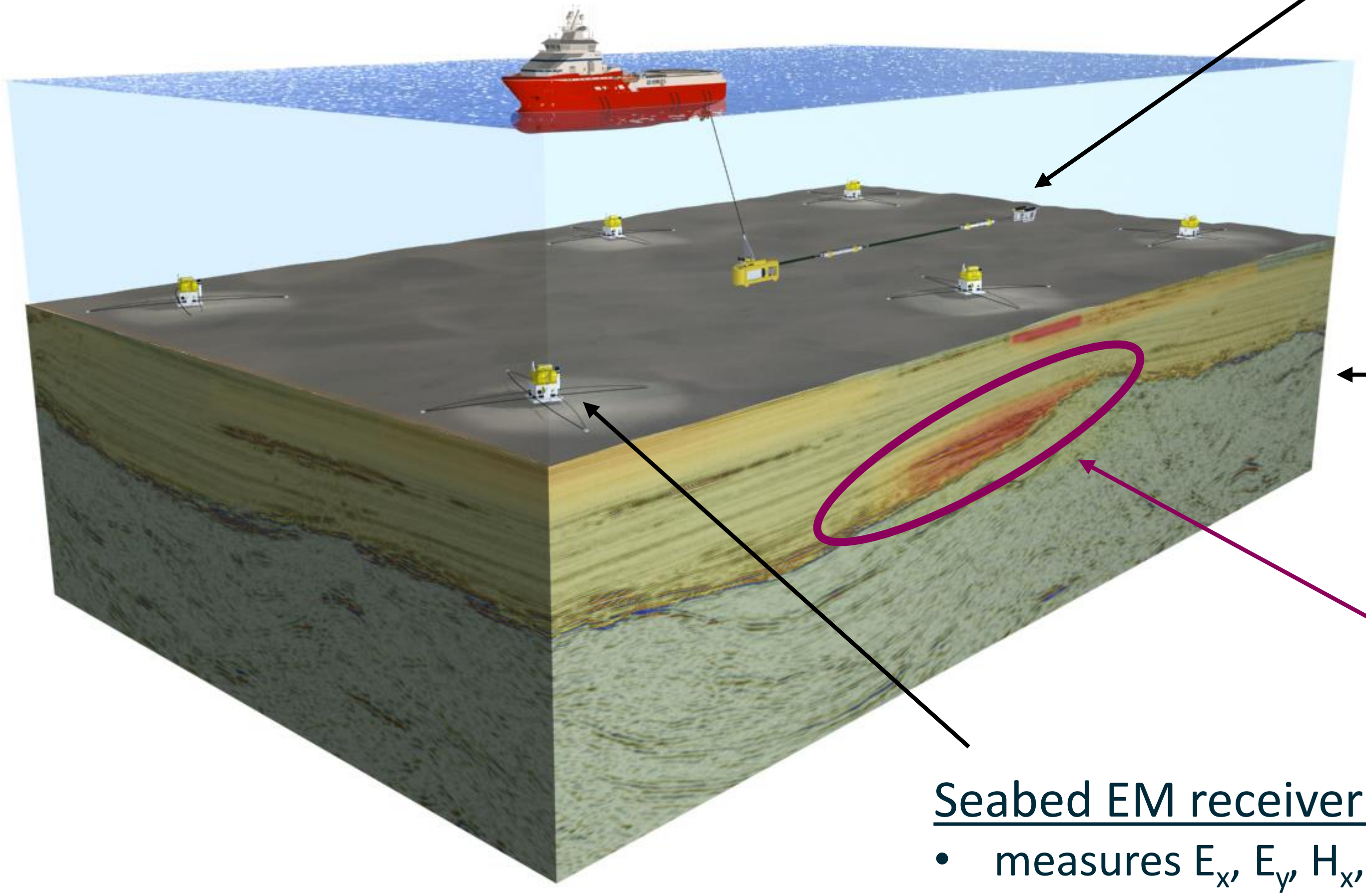
Large-scale 3-D EM modelling with a Block Low-Rank multifrontal direct solver

Daniil V. Shantsev,[1,2] Piyoosh Jaysaval,[2,3] Sébastien de la Kethulle de Ryhove,[1]
Patrick R. Amestoy,[4] Alfredo Buttari,[5] Jean-Yves L'Excellent[6] and Theo Mary[7]

Outline

- Marine Controlled-source EM (CSEM) method
- CSEM forward & inverse problems
- MUMPS-BLR for CSEM problems
- Air effects
- Conclusions

Marine CSEM survey



<u>Horizontal electric dipole source:</u>

- current up to 7200 Amp
- frequency up to ~20 Hz
- tow-distance ~1,000 km i.e. ~10,000 source positions

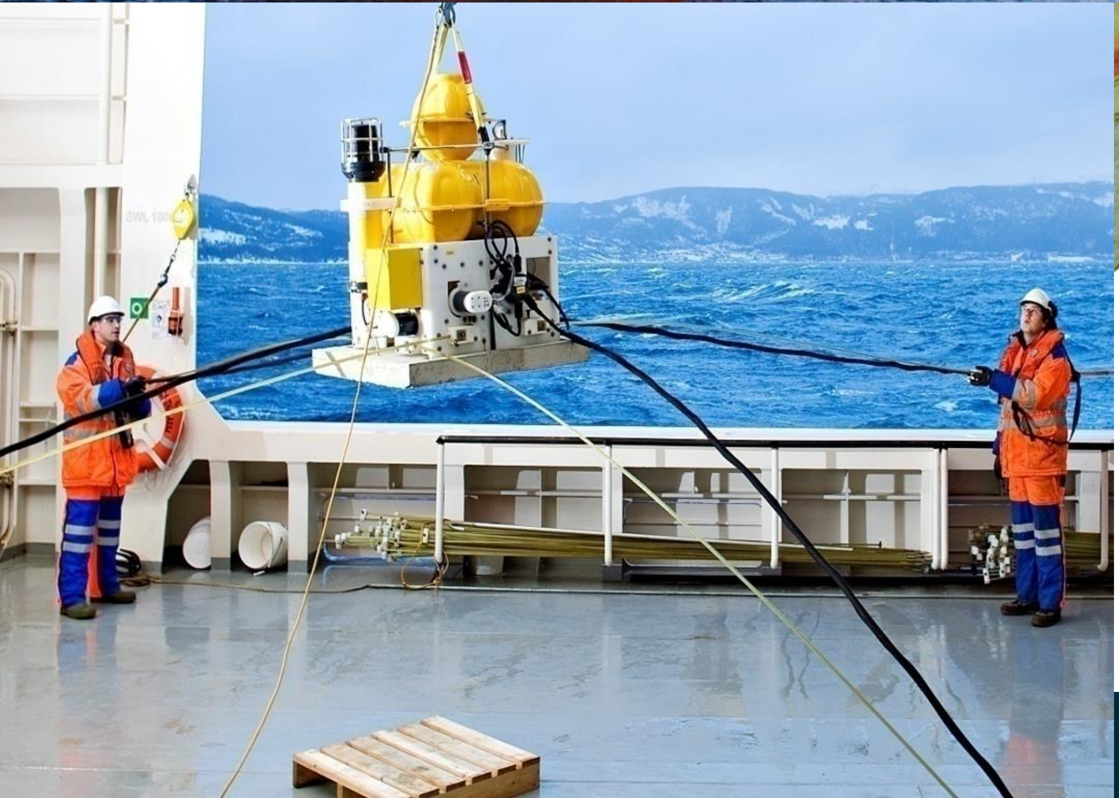
3D subsurface resistivity distribution:

- inverted from the measured data

High-resistive anomaly indicating a possible oil/gas accumulation: impact on drilling decision

<u>Seabed EM receiver:</u>

- measures E_x, E_y, H_x, H_y
- 2D grid, typically ~100 receivers

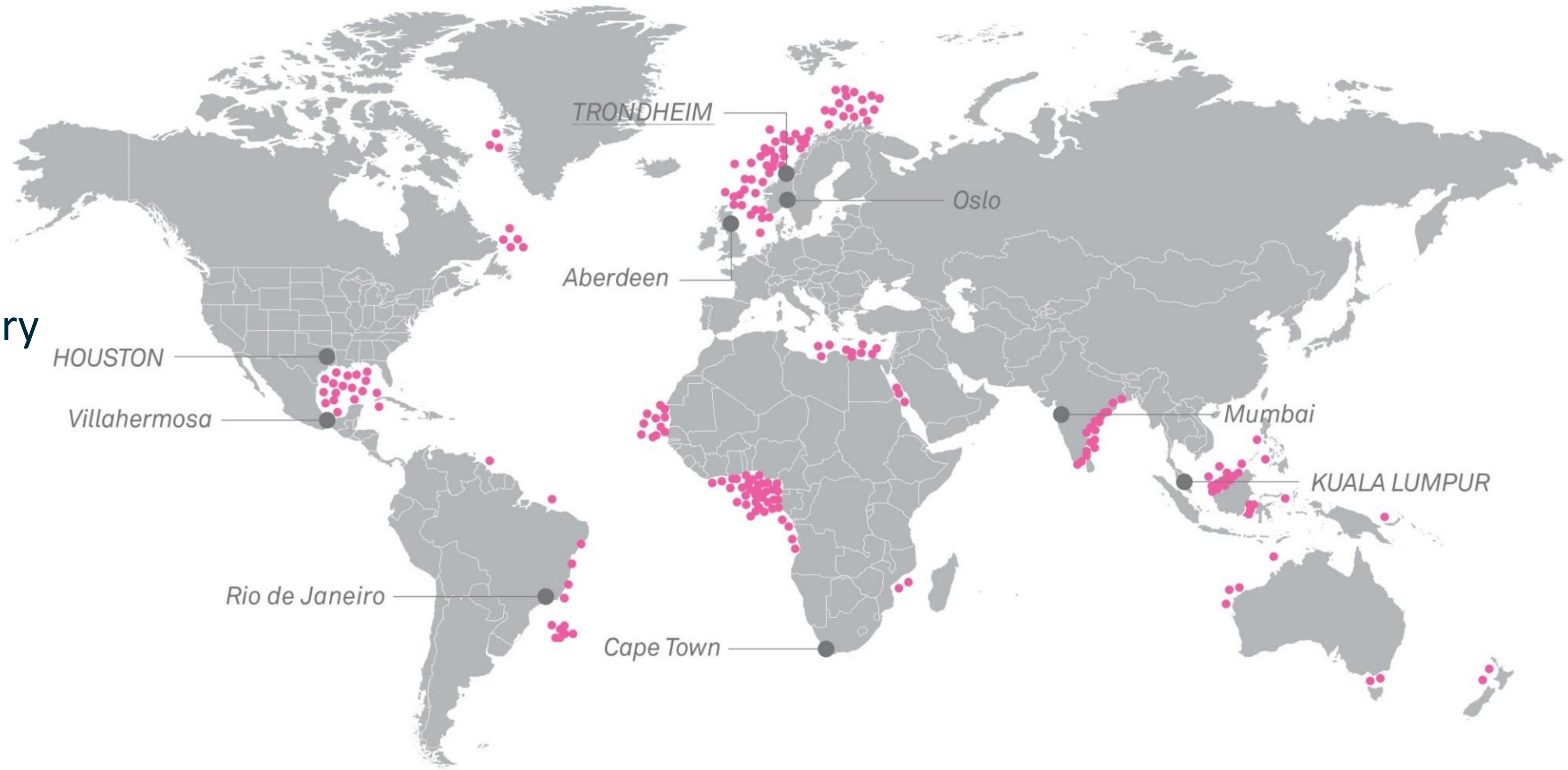


Spot the difference.

EMGS

- 2 – 4 vessels
- ~ 200 employees
- ~1000 surveys
- ~ 100,000 km^2 data library
  (10-20% area of France)

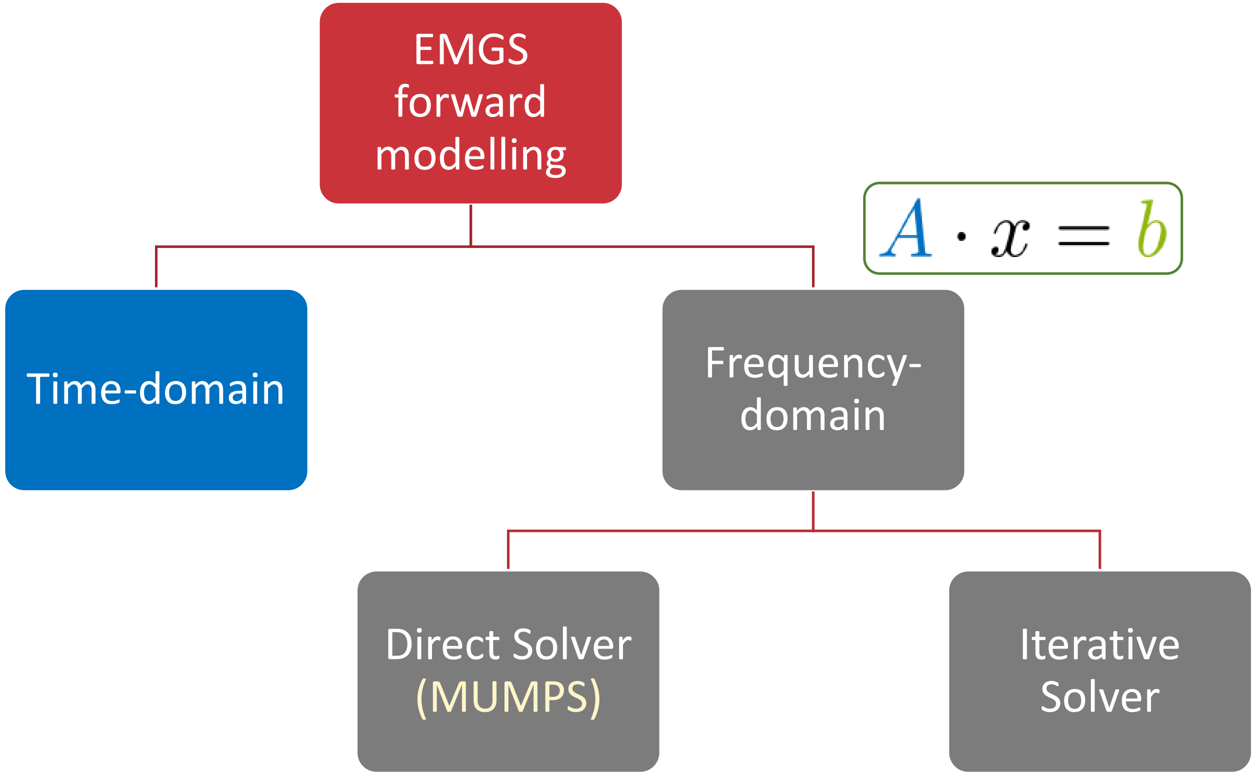
- R&D group: ~10 people



CSEM Forward & Inverse problems

Spot the difference.

Forward problem



Can be the preferred choice
for very large number of RHSs

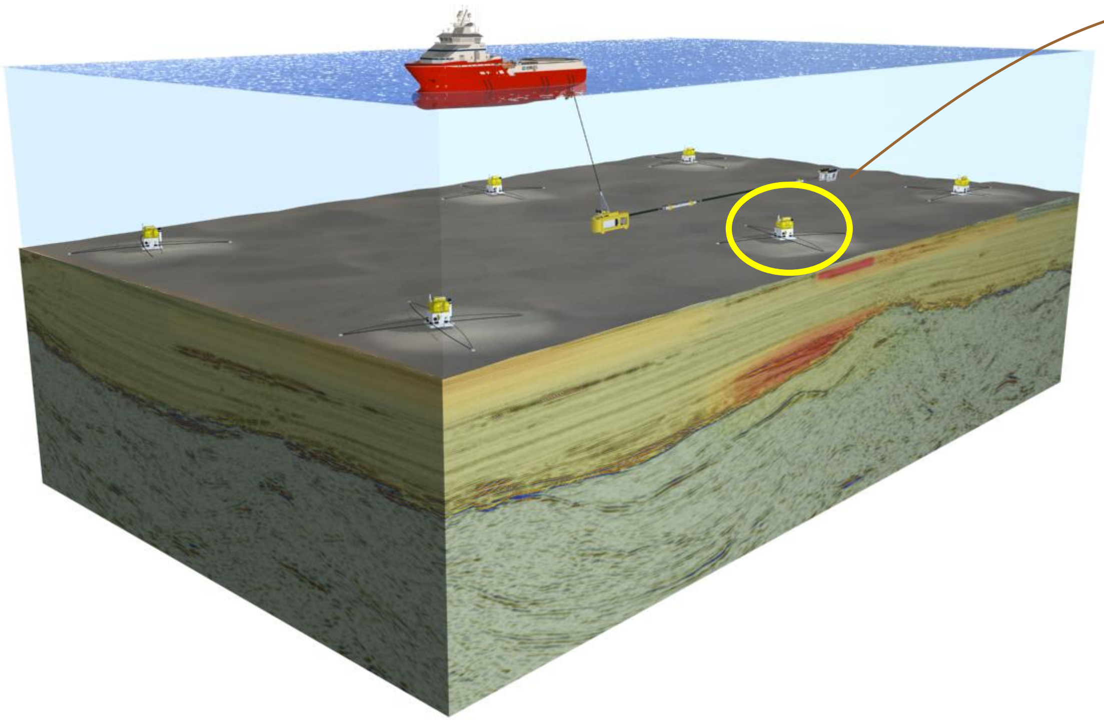
Equation to solve:
(without displacement current)

$$\nabla \times \nabla \times \mathbf{E} - i\omega\mu_0\sigma\mathbf{E} = i\omega\mu_0 J_{\text{source}}$$

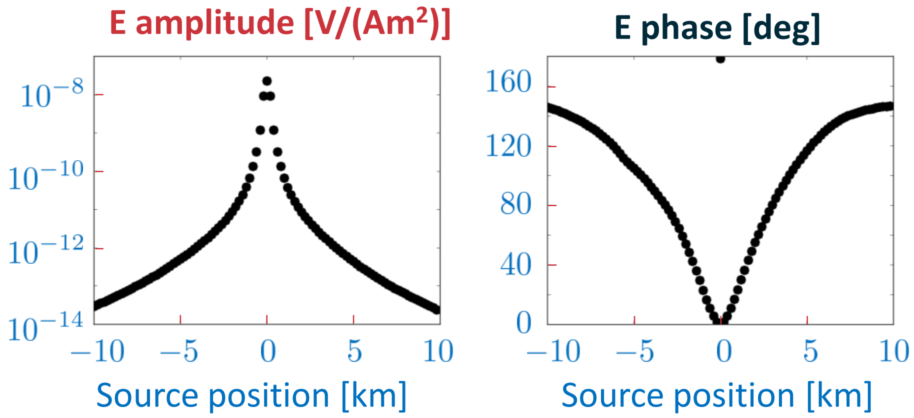
This study:

- VTI
- Finite-difference based on Yee grid
- Unknowns: E_x, E_y, E_z
- 13 non-zero elements in each row of A
- Symmetric A

CSEM data



Data from one receiver at one frequency



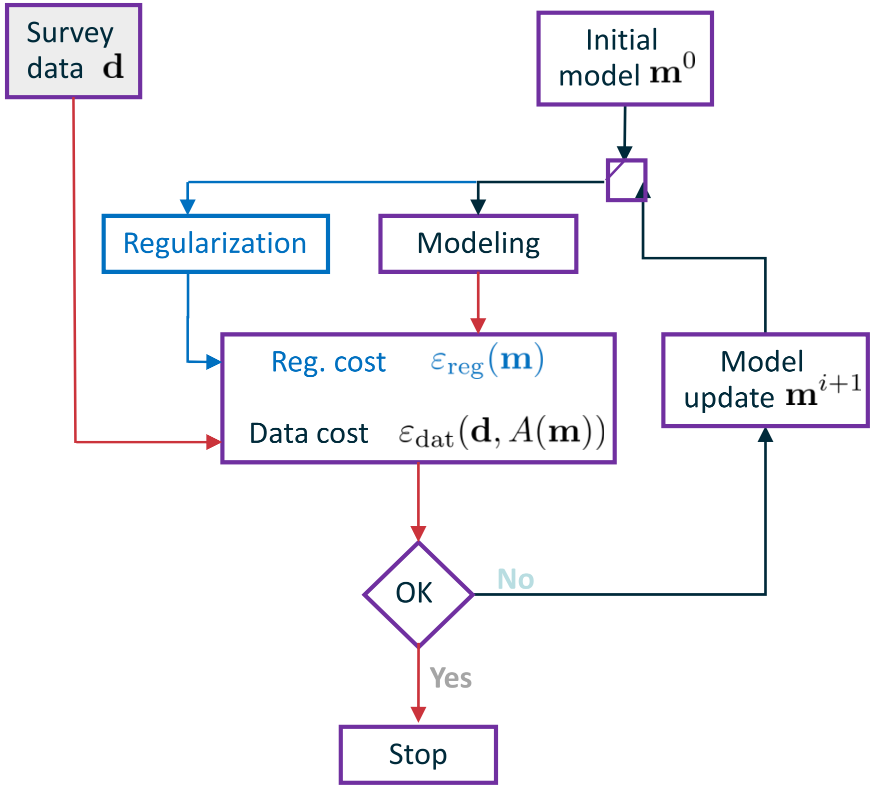
Number of receivers: $N_r \sim 100$

Number of source positions: $N_s \sim 10{,}000$

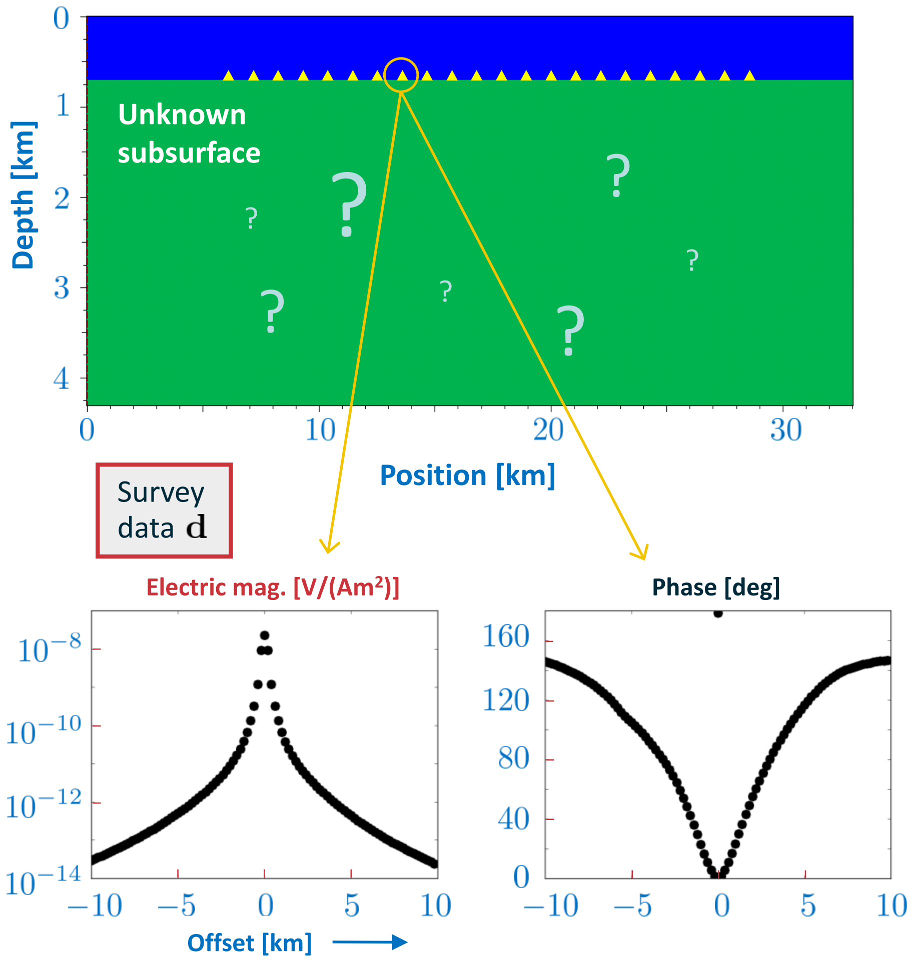
- ~200 datapoints per line (for 100 m sampling, 20 km line)
- ~5 source lines for each receiver
- Amplitude + phase
- 3 – 5 frequencies
- 2 – 4 field components (E_x,E_y)
- ~100 receivers
- **Total: a few millions of data samples**

Inversion algorithm

Inversion algorithm

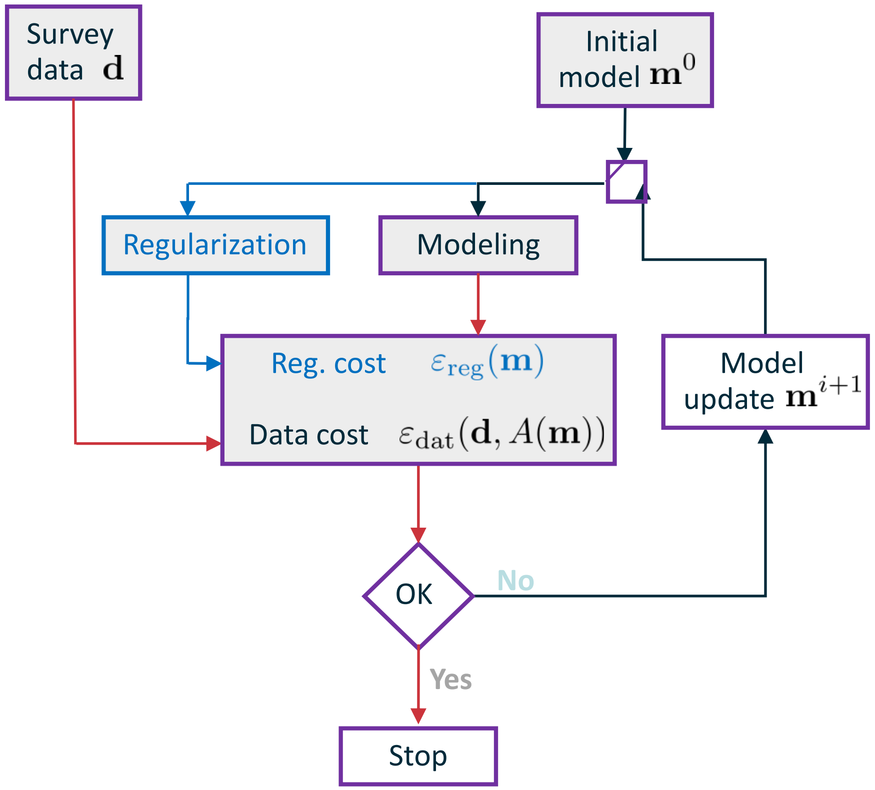


Inversion example

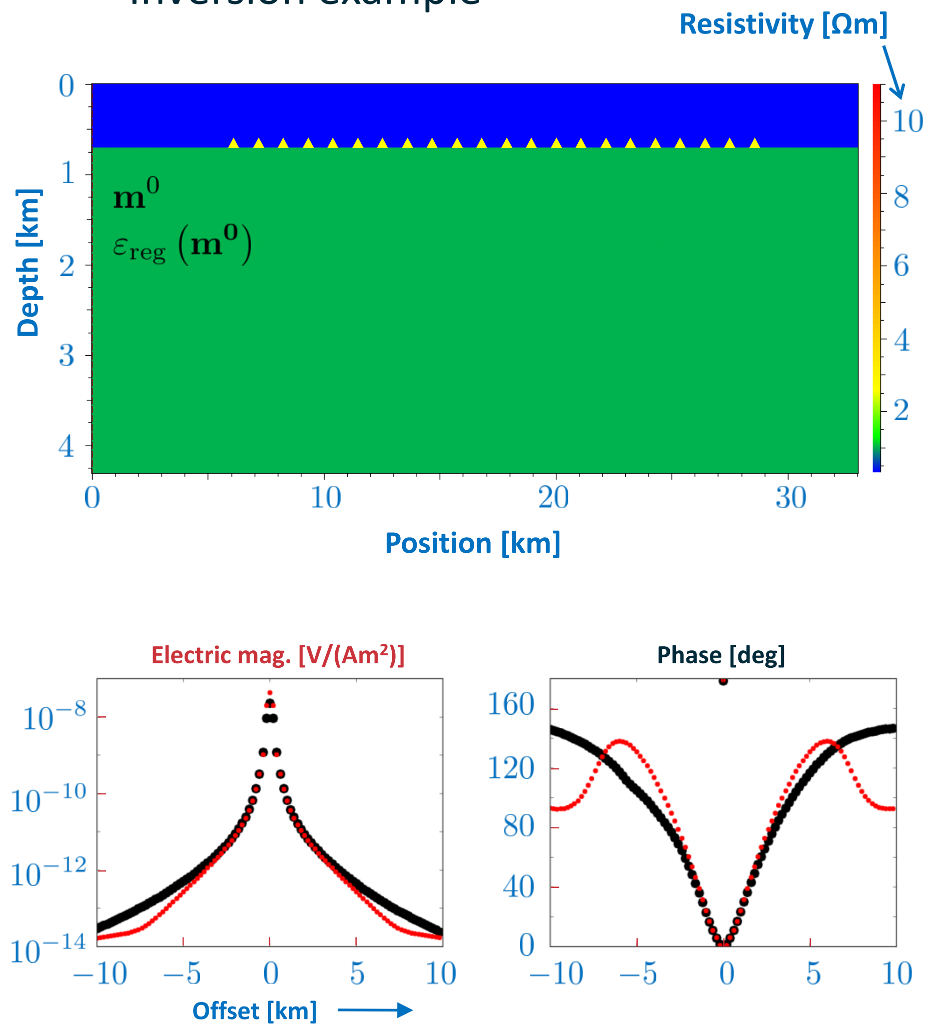


Inversion algorithm

Inversion algorithm

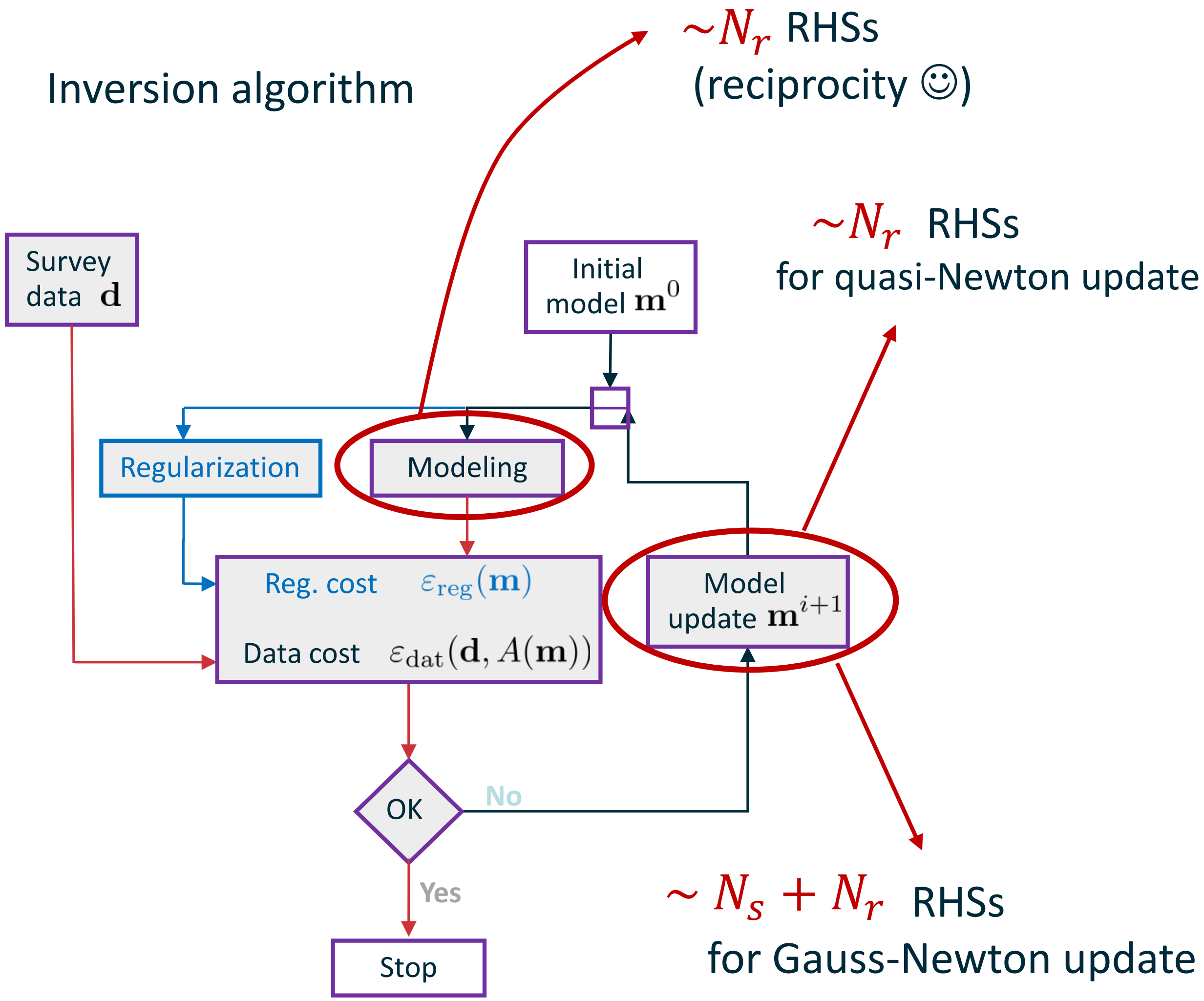


Inversion example

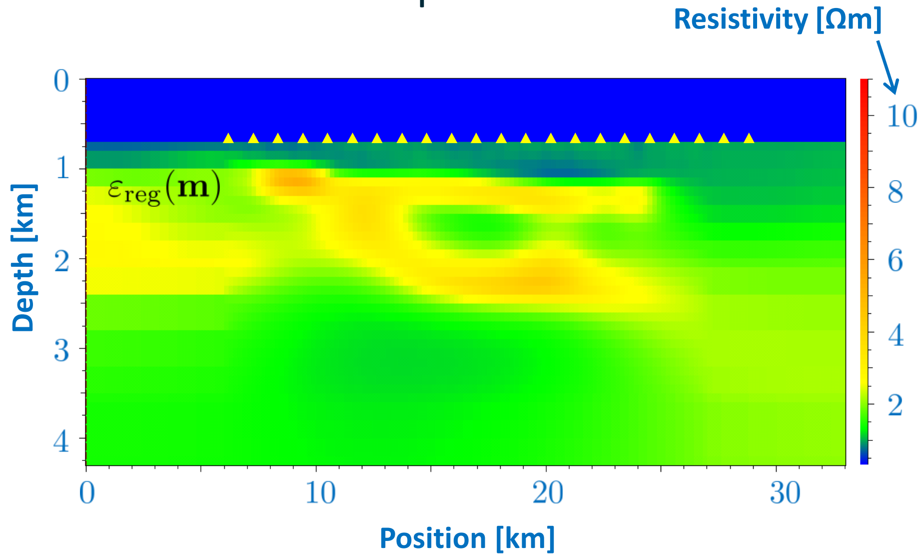


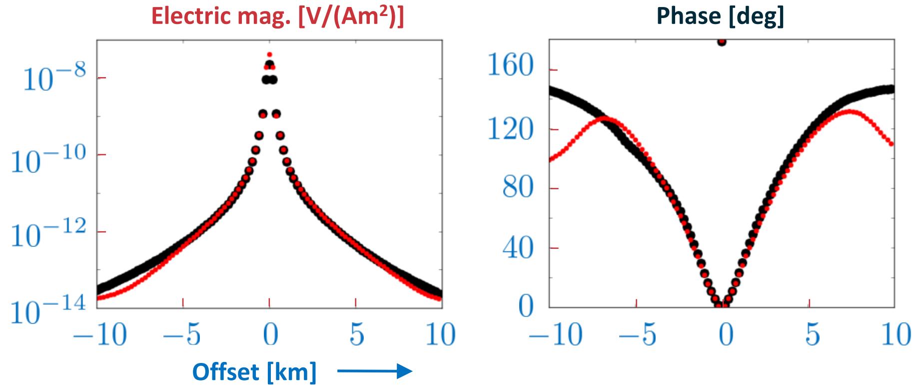
Inversion algorithm

Inversion algorithm



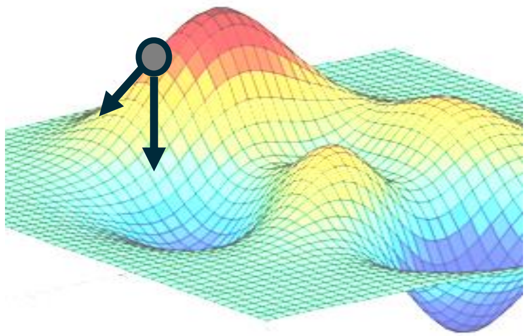
Inversion example





Quasi-Newton and Gauss-Newton inversion

- Cost function to be minimized: $\varepsilon(\mathbf{m}) = \varepsilon_{\text{data}}(\mathbf{m}) + \lambda\, \varepsilon_{\text{reg}}(\mathbf{m}) = \sum_{k=1}^{N} r_k r_k^* + \lambda\, \varepsilon_{\text{reg}}(\mathbf{m})$
- Model update $\mathbf{\Delta m}$ is found from: $(H_{\text{data}} + \lambda H_{\text{reg}})\mathbf{\Delta m} = -\,\mathbf{g}$
- The Hessian matrix is $H_{\text{data}} \approx J^{\dagger} J + \text{c.c.}$



Gradient vector

$$\mathbf{g} = \begin{pmatrix} \dfrac{\partial \varepsilon}{\partial \sigma_1} \\ \vdots \\ \dfrac{\partial \varepsilon}{\partial \sigma_M} \end{pmatrix}$$

Jacobian (sensitivity) matrix

$$J = \begin{pmatrix} \dfrac{\partial r_1}{\partial \sigma_1} & \cdots & \dfrac{\partial r_1}{\partial \sigma_M} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial r_N}{\partial \sigma_1} & \cdots & \dfrac{\partial r_N}{\partial \sigma_M} \end{pmatrix}$$

Number of data samples, $\sim N_r \times N_s$

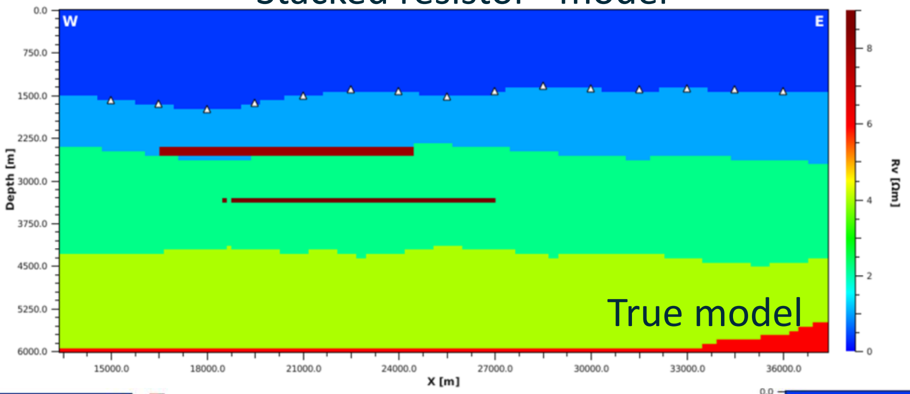
Number of model parameters

	Hessian $\boldsymbol{H}$	Number of RHSs in forward problem
Quasi-Newton (e.g. BFGS)	approximated using successive gradients $\mathbf{g}$	$\sim N_r$ ~ 100
Gauss-Newton	computed from Jacobian $H_{\text{data}} \approx J^{\dagger} J + \text{c.c.}$	$\sim N_s + N_r$ $\sim 10{,}000$

Gauss-Newton is better

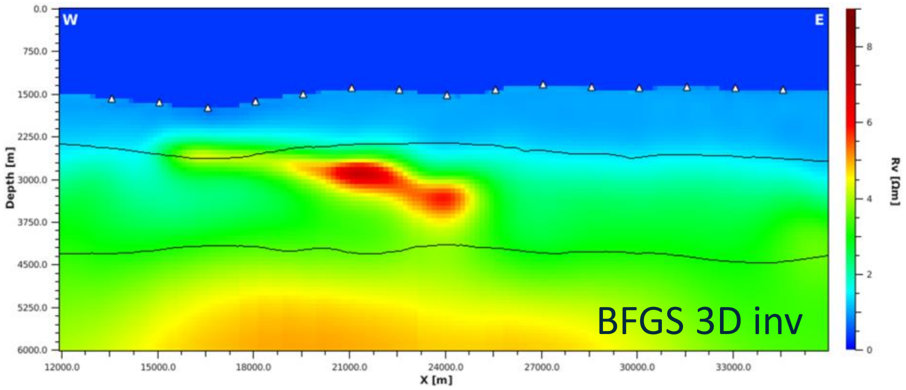
- Gauss - Newton is more expensive, but a much powerful method
- It will take over in the future:
    - 2008: Launch BFGS
    - 2016: Launch Gauss-Newton

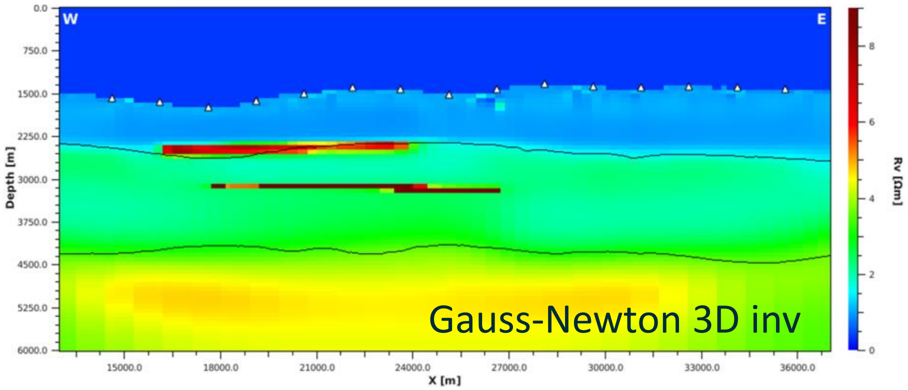
“Stacked resistor” model



SEG 2016:

Comparing large-scale 3D Gauss–Newton and BFGS CSEM inversions. Anh Kiet Nguyen, Janniche Iren Nordskag, Torgeir Wiik, Astrid Kornberg Bjørke, Linus Boman, Ole Martin Pedersen, Joseph Ribaudo, and Rune Mittet (2016) Comparing large-scale 3D Gauss–Newton and BFGS CSEM inversions. SEG Technical Program Expanded Abstracts 2016: pp. 872-877. doi: 10.1190/segam2016-13858633.1





emgs

MUMPS for CSEM

Spot the difference.

MUMPS for CSEM: previous studies

		Number of unknowns
Streich	Geophysics 2009	0.9 millions
da Silva et al.	Geophysics 2012	4.2 millions
Puzyrev et al.	Comp & Geos. 2016	7.8 millions

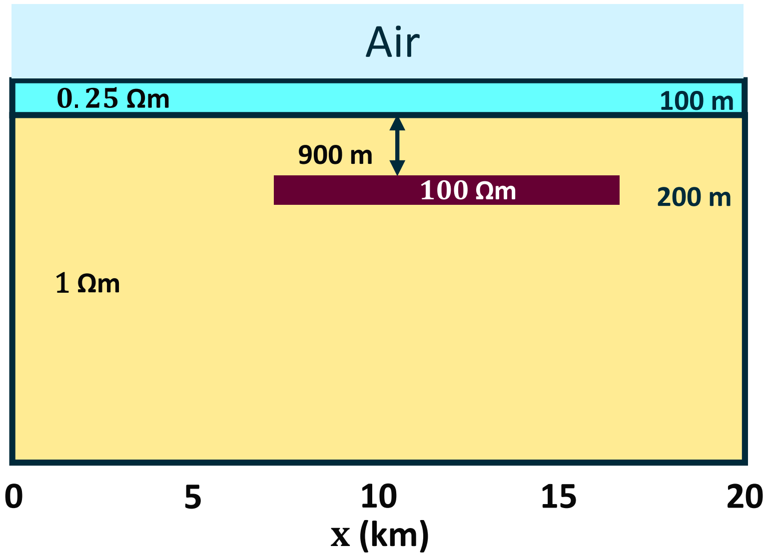
Goals of the present study :

- Use **BLR** for factorization of CSEM matrices
- Test problems with **>20 millions** unknowns
- Compare MUMPS vs Iterative solver

Models and Matrices

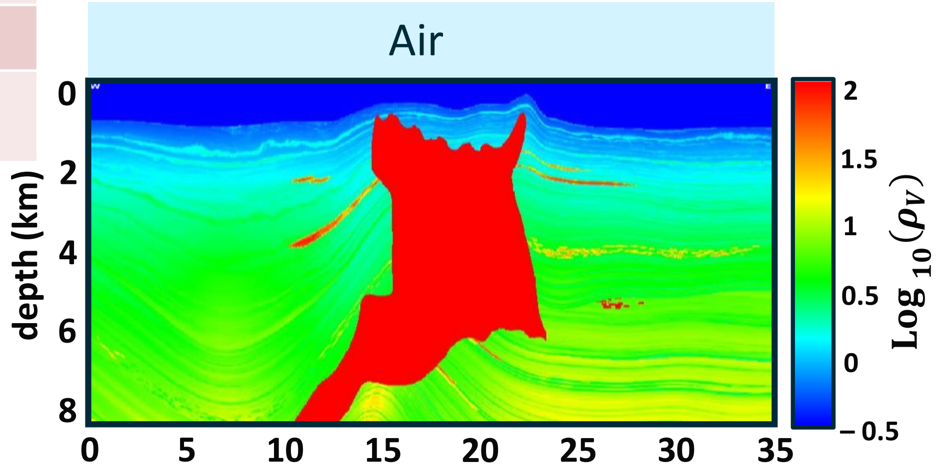
Half-space + Target H-model							
Grid	**Matrix**	$\mathbf{dx = dy}$	$\mathbf{dz}$	$\mathbf{N_x = N_y}$	$\mathbf{N_z}$	**Number of unknowns**	**Number of non-zero elements**
$G1$	**H1**	400	200	64	74	909,312	11,658644
$G2$	**H3 / D3**	200	200	114	74	2,885,112	37,148,644
$G3$	**H17**	100	100	214	127	17,448,276	225,626,874

SEAM S-model								
Grid	**Matrix**	$\mathbf{dx = dy}$	$\mathbf{dz}$	$\mathbf{N_x}$	$\mathbf{N_y}$	$\mathbf{N_z}$	**Number of unknowns**	**Number of non-zero elements**
$G4$	**S3**	480	80	98	87	130	3,325,140	42,836,538
$G5$	**S21**	240	40	181	160	237	20,590,560	266,361,112



SEAM model:

- Created by SEG Advanced Modelling
- Salt body
- Representative for Gulf of Mexico

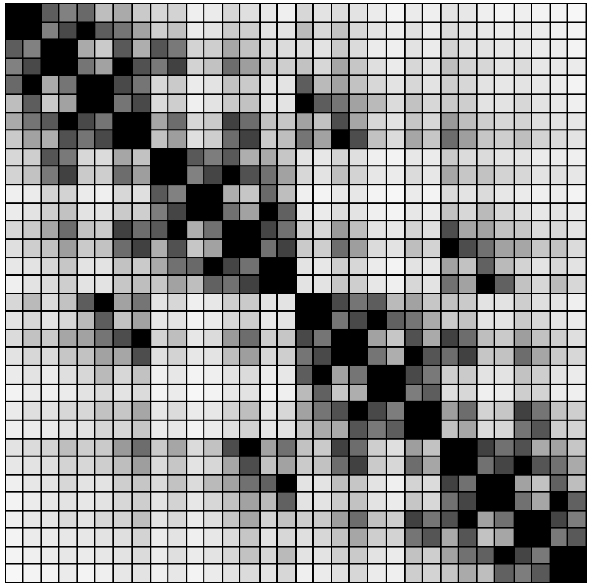


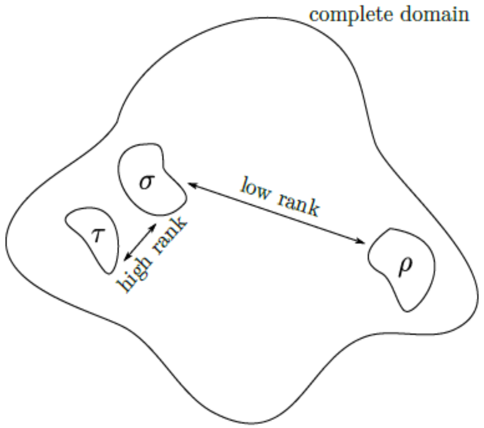
Block-low-rank (BLR) algorithm

Input: a symmetric matrix $\mathbf{A}$ of $p \times p$ blocks
Output: the factors matrices $\mathbf{L}$, $\mathbf{D}$

for $k = 1$ **to** p **do**
  Factor: $\mathbf{L_{kk} D_{kk} L_{kk}^T} = \mathbf{A_{kk}}$
  for $i = k+1$ **to** p **do**
    Solve: $\mathbf{L_{ik}} = \mathbf{A_{ik} L_{kk}^{-T} D_{kk}^{-1}}$
    Compress: $\mathbf{L_{ik}} \approx \mathbf{Y_{ik} Z_{ik}^T}$
    for $j = k+1$ **to** i **do**
      Update: $\mathbf{A_{ij}} = \mathbf{A_{ij}} - \mathbf{L_{ik} D_{kk} L_{jk}^T}$
      $\approx \mathbf{A_{ij}} - \mathbf{Y_{ik}(Z_{ik}^T D_{kk} Z_{jk}) Y_{jk}^T}$
    end for
  end for
end for

- BLR format is used to compress fronts
- The compression accuracy is controlled by the BLR threshold ε that varied from 10^{-4} to 10^{-16}
- Block size: 256
  (or 416 for largest matrix, S21)

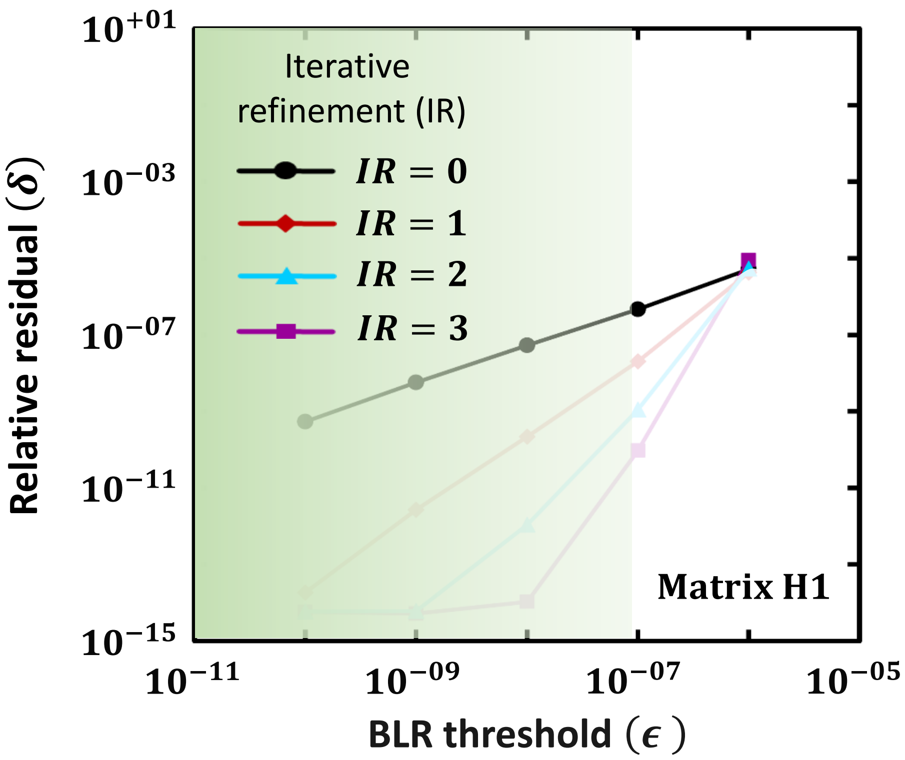




BLR-compressed matrix structure.
Block darkness = Compression rate
Figures from Amestoy et al. 2015

BLR threshold & Solution accuracy

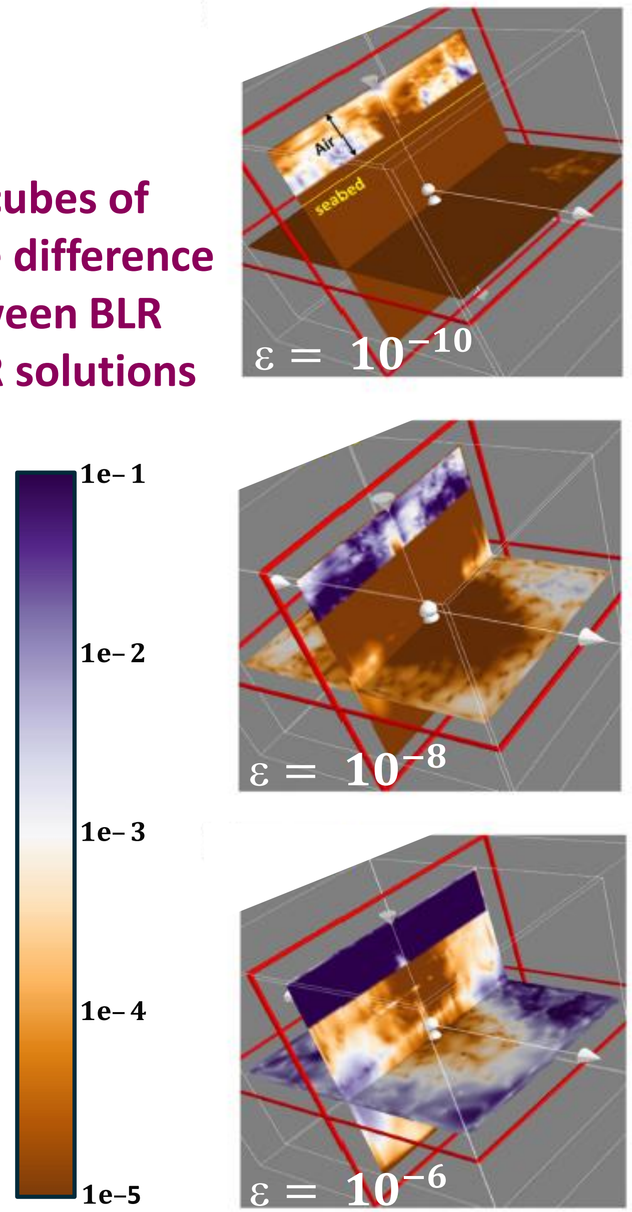
Relative residual L2 norm: $\delta = \frac{||\mathbf{s} - \mathbf{M}\mathbf{x}^{\epsilon}||}{||\mathbf{s}||} < 10^{-6}$



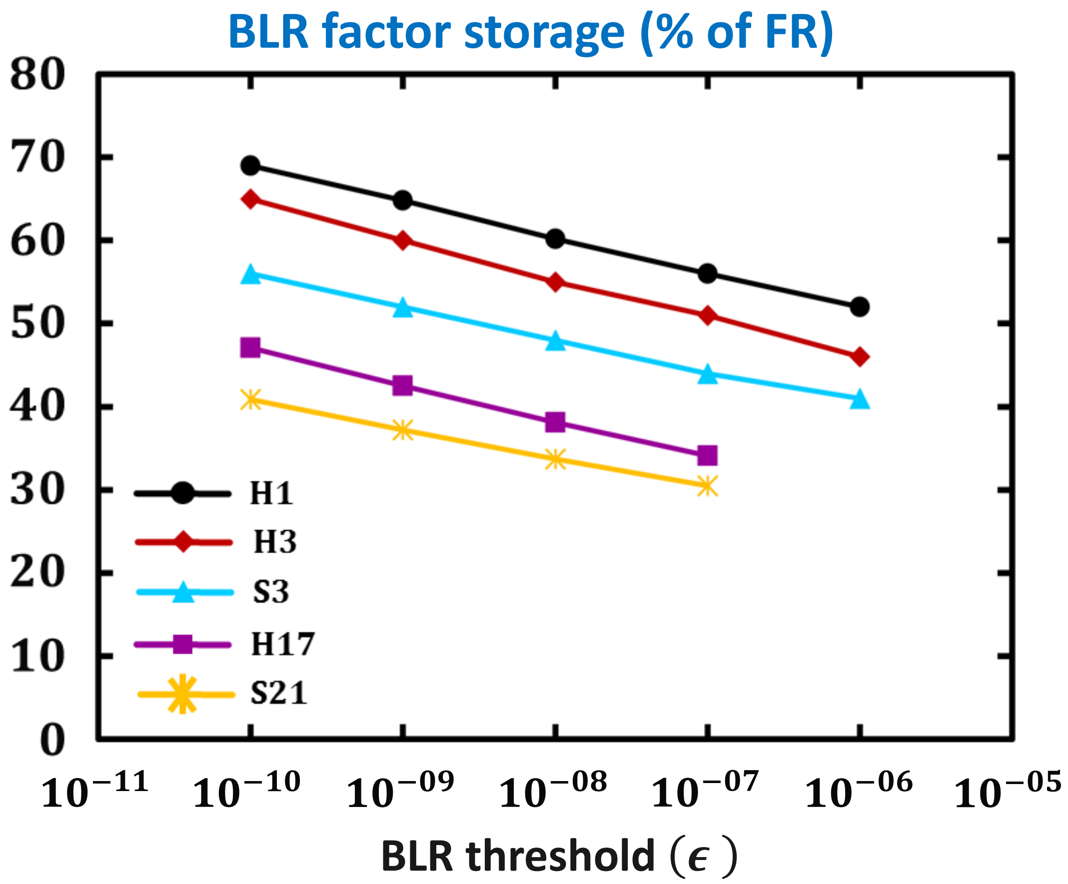
Optimal BLR threshold:

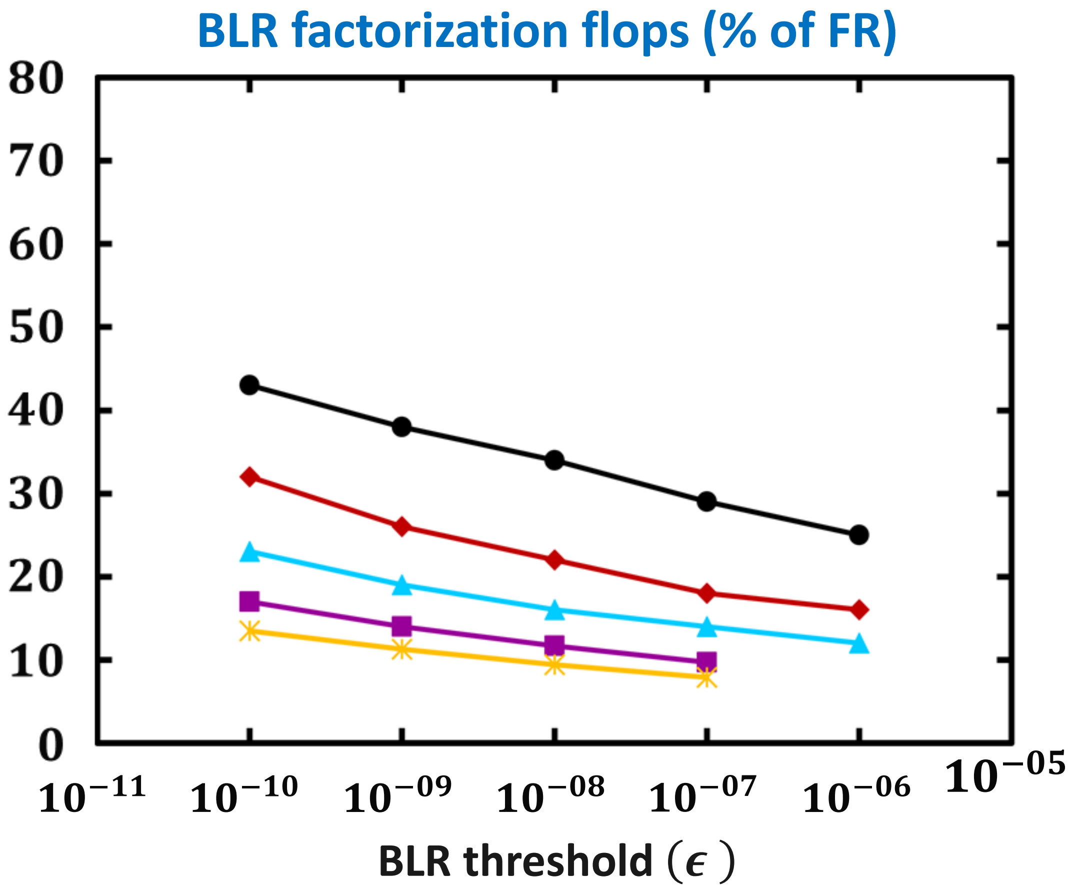
$\varepsilon = 10^{-7}$

3D cubes of relative difference between BLR and FR solutions



BLR savings

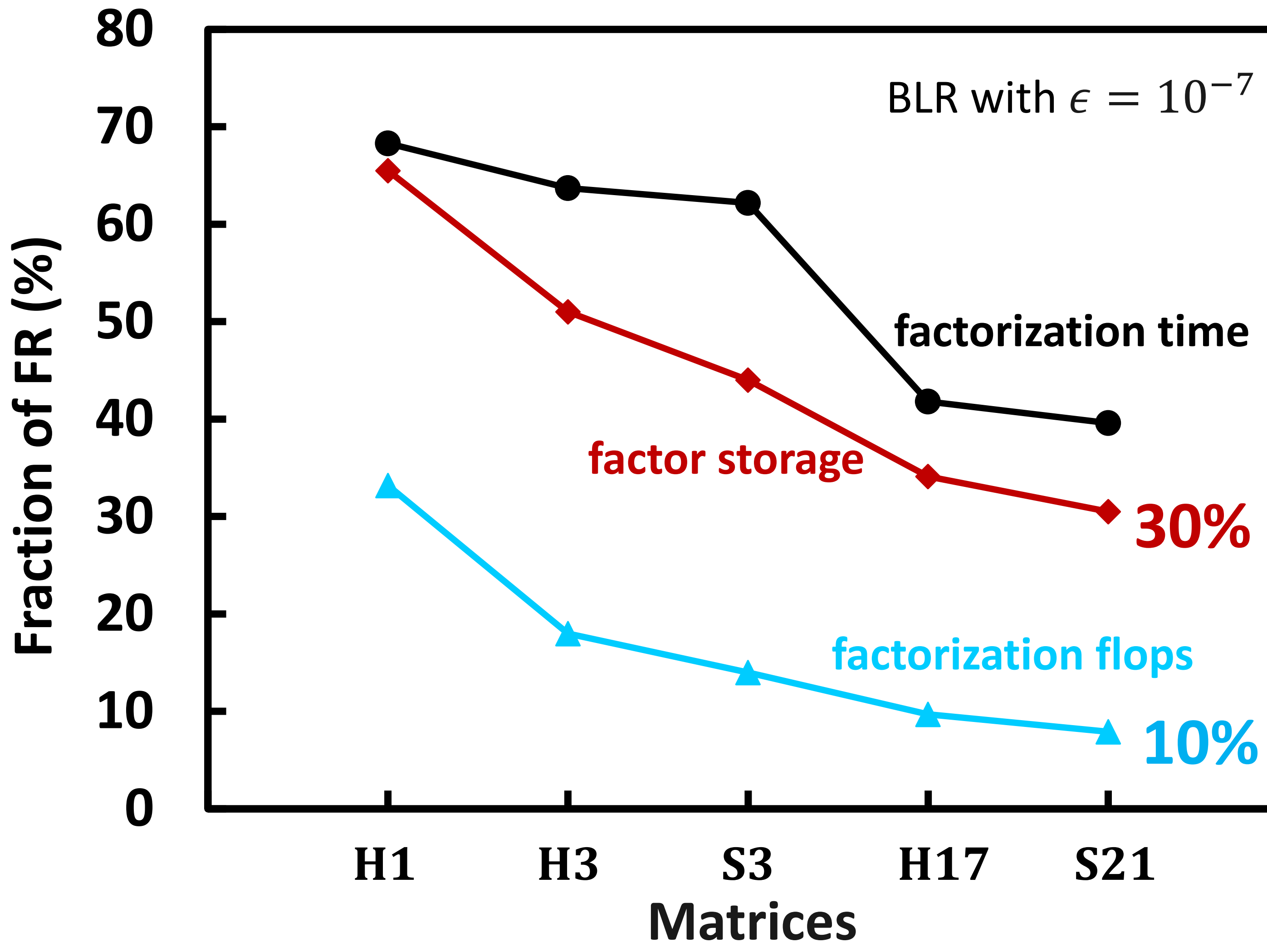




Weak dependence:

- *Choice of the BLR threshold is not critical*
- *Large gains even for strict accuracy requirements*

BLR savings



Hardware:

- EOS supercomputer
- 90 MPI tasks × 10 threads

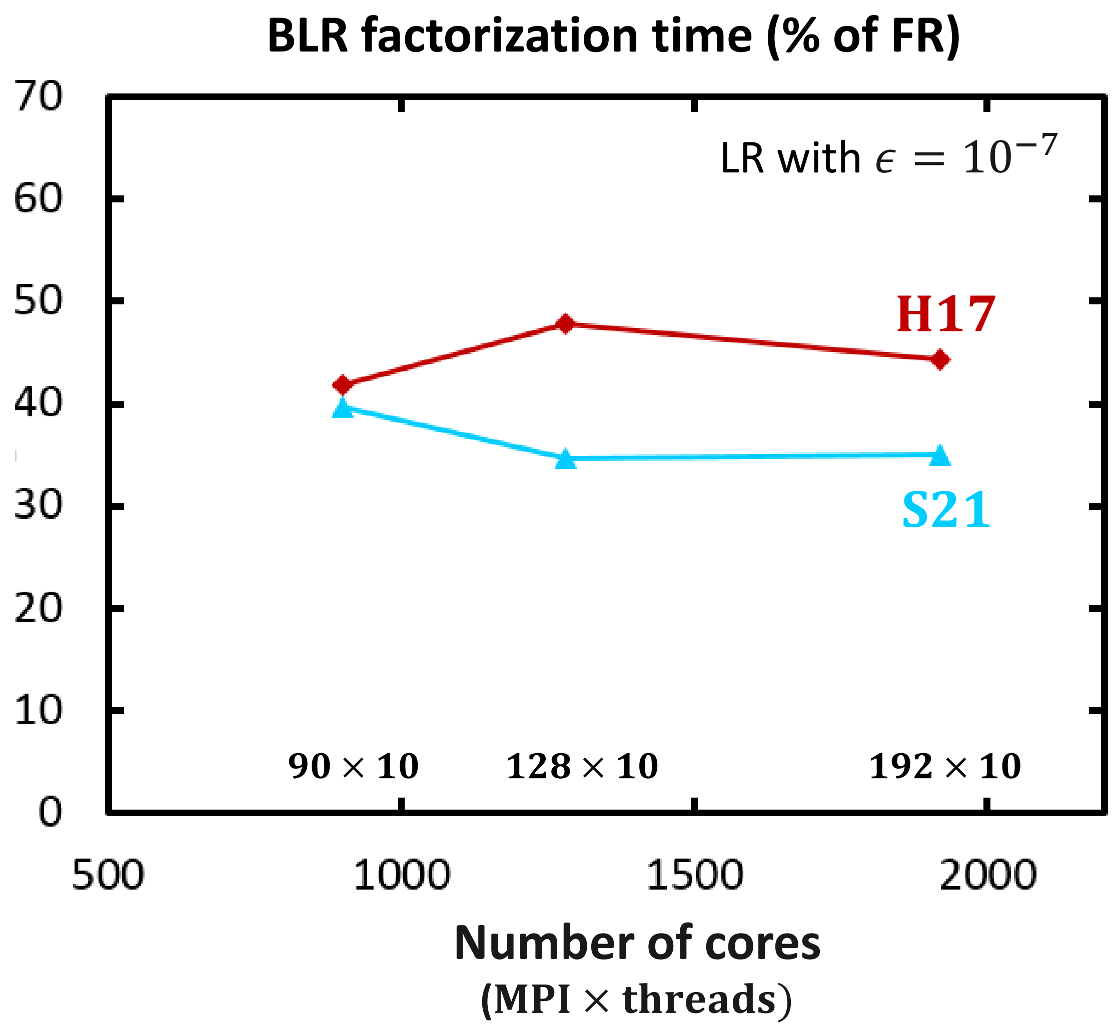
Hardware details:

CALMIP supercomputer EOS – a BULLx DLC system, 612 nodes, each composed of two Intel Ivybridge processors with 10 cores (total 12 240 cores) running at 2.8 GHz per node and 64 GB/node, https://www.calmip.univ-toulouse.fr/

NB:

Memory reduction due to storage savings has not yet been implemented for these tests, hence, the potential gains in run-time are even larger

Scalability



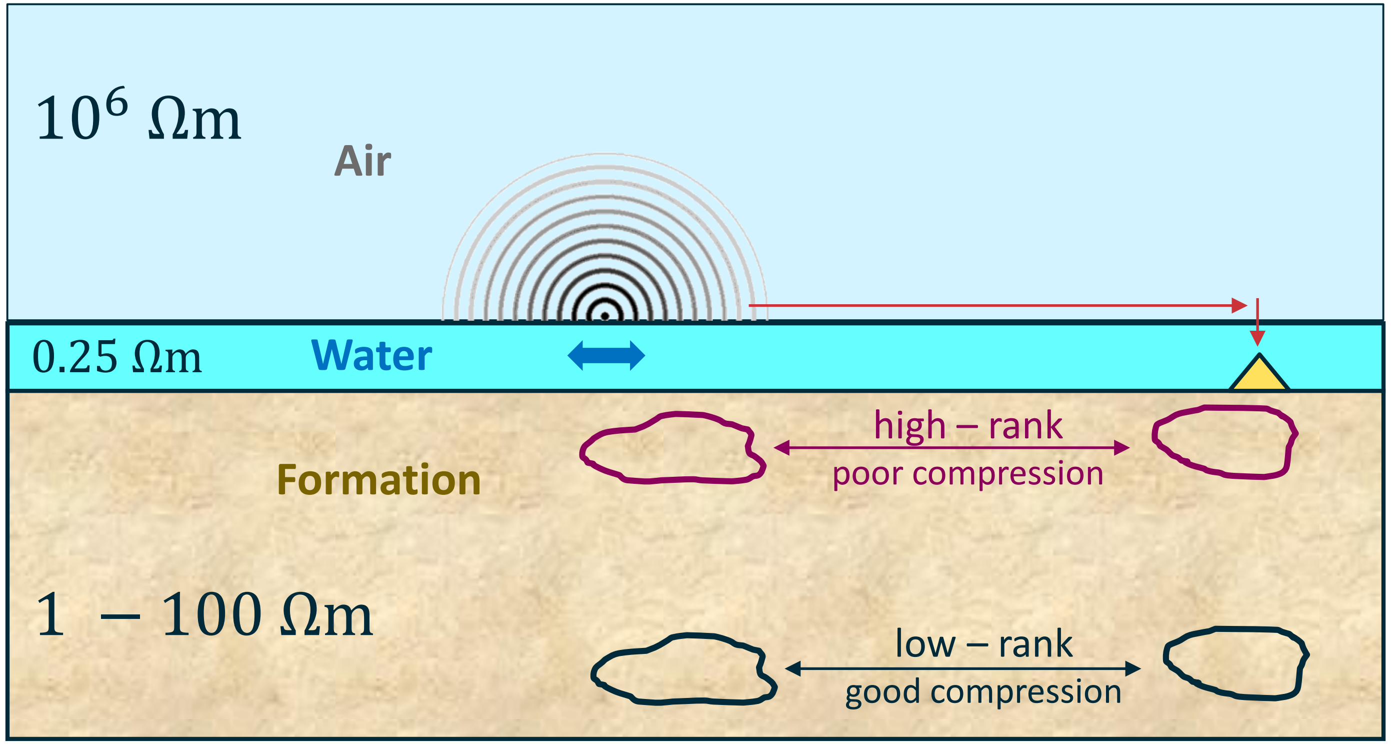
Robust BLR-gains independent of the number of cores

emgs

Air effects

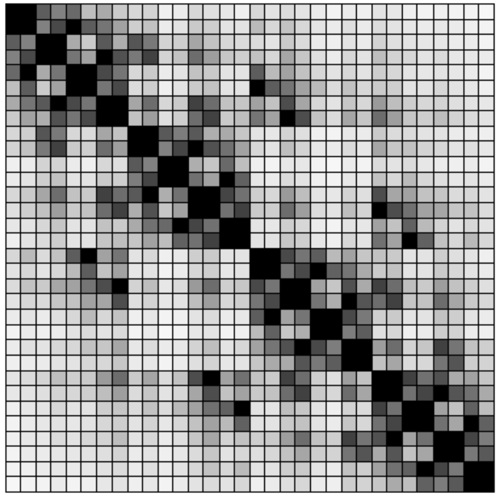
Spot the difference.

Air-wave



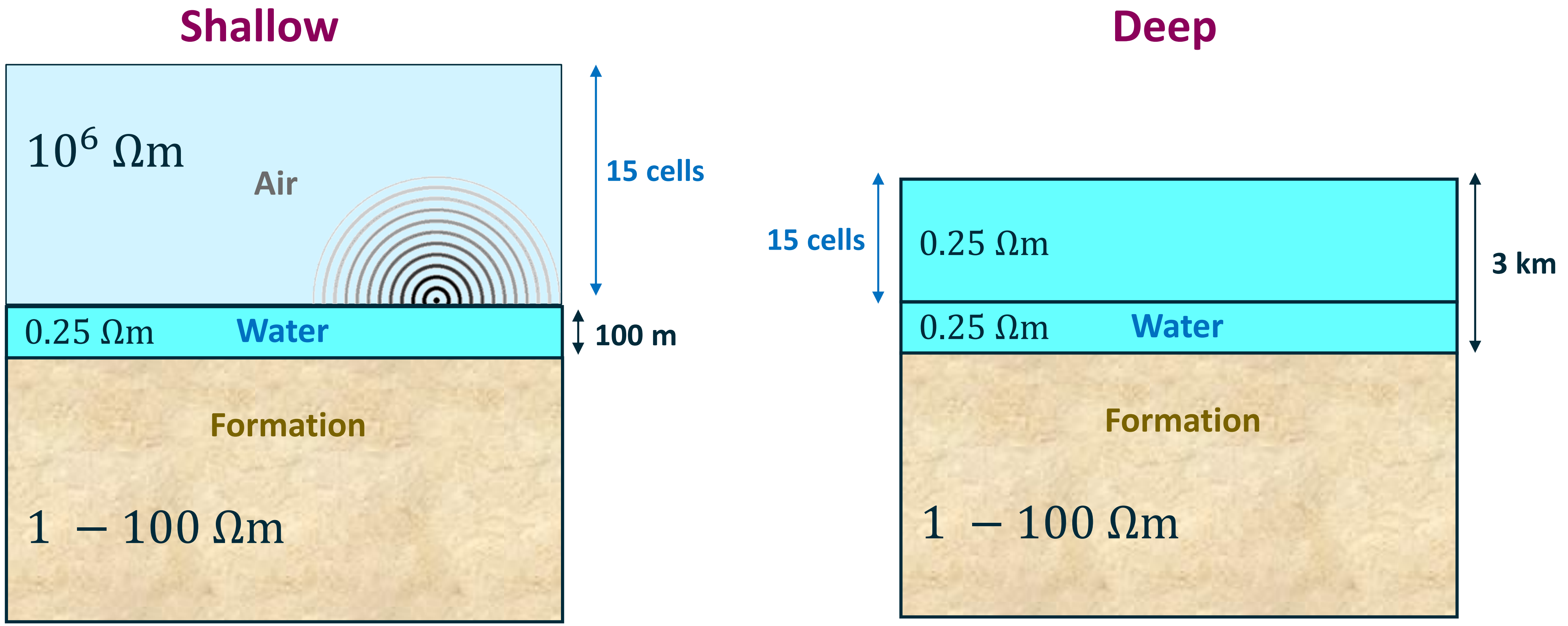
In marine CSEM variations of properties (resistivity is extreme)

- Air has almost infinite resistivity
- EM waves propagate there fast (speed of light) and without attenuation
- Air effectively connects distant parts of the model



BLR- compressed matrix structure.
Figure from Amestoy et al. 2015

Shallow-water & Deep-water models

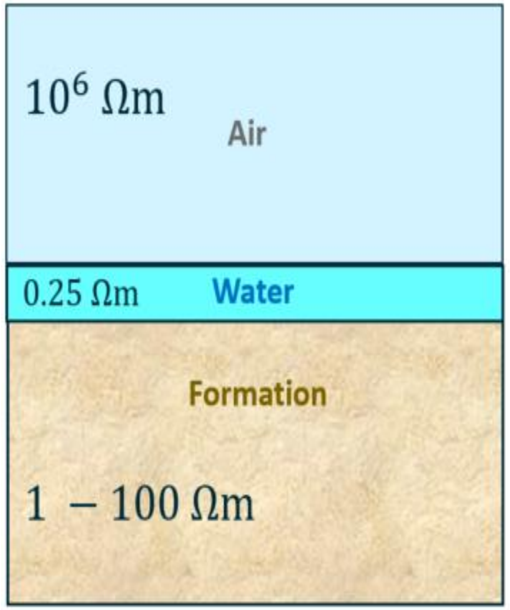


The number of cells is the same for both models,
i.e. the matrices have identical structure

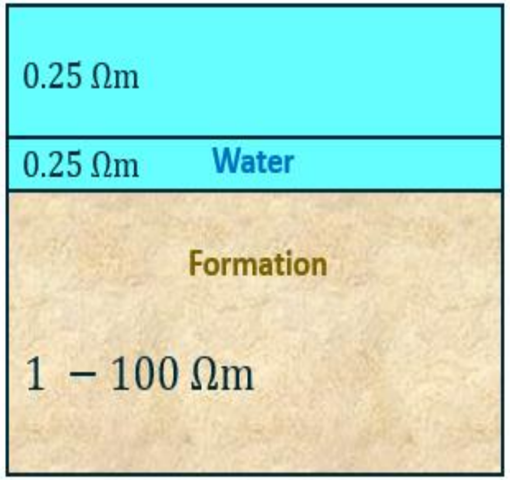
Matrix size: $4.9 \cdot 10^6$

BLR & Full Rank (FR) flops

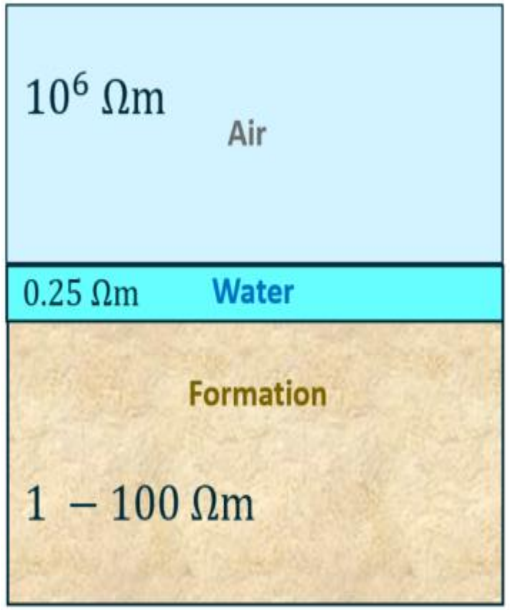
FR - Shallow



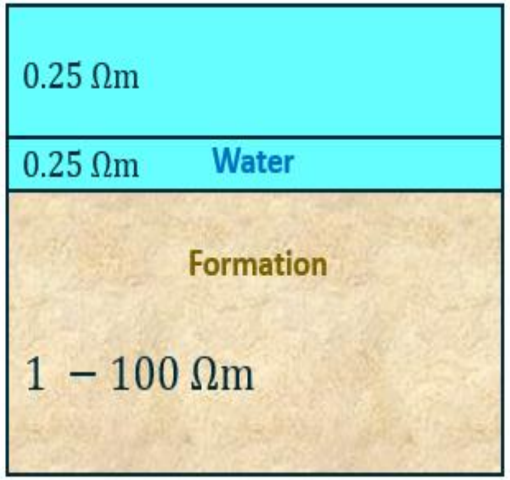
FR - Deep



BLR - Shallow



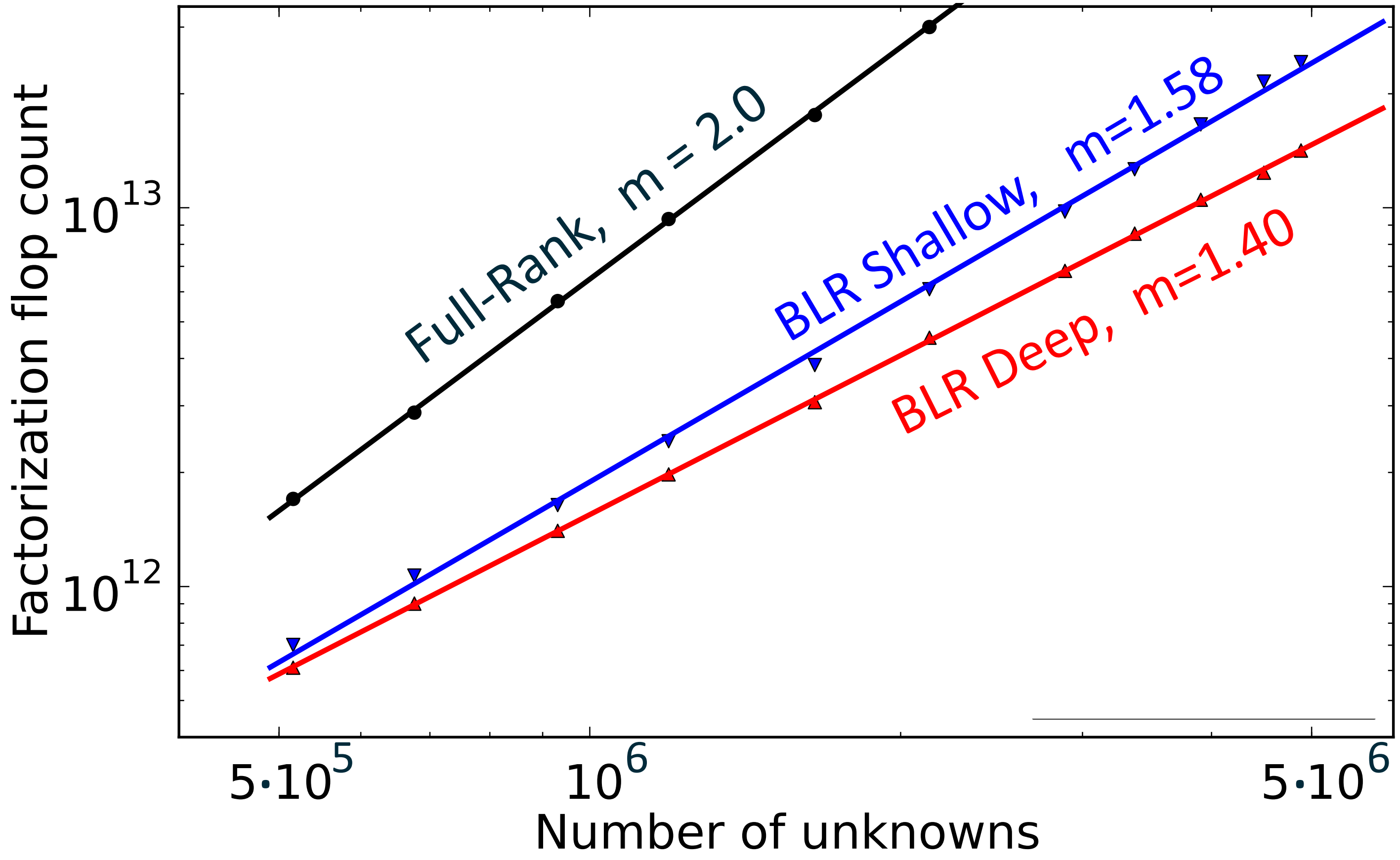
BLR - Deep



$162 \cdot 10^{12}$ $162 \cdot 10^{12}$ $24 \cdot 10^{12}$ $14 \cdot 10^{12}$

100 % **15 %** **9 %**

Flops complexity



Number of flops for matrix factorization

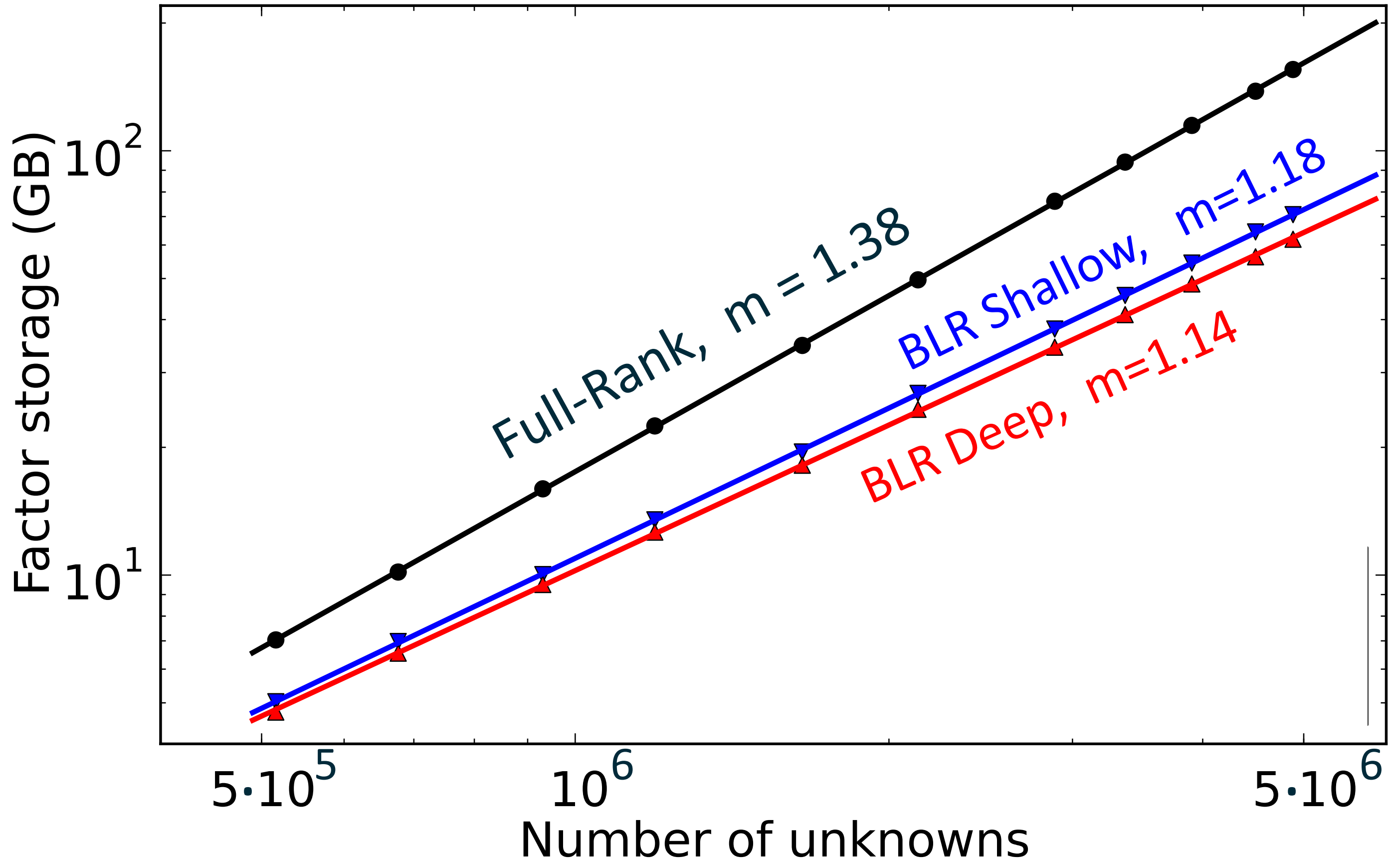
Matrix size (number of unknowns)

$$N_{flops} \propto N^m$$

Consistent with (or better than):

- Theory: $m = 1.7$
  Amestoy et al. SIAM J. Sci. Comput 2017
- 3D Seismic: $m = 1.78$
  Amestoy et al. SIAM J. Sci. Comput 2015

Factor storage complexity



Memory needed to store factors

Matrix size (number of unknowns)

$$N_{bytes} \propto N^m$$

Consistent with:

- Theory:
  $m = 1.33$ (FR), $m = 1.17$ (BLR)
  Amestoy et al. SIAM J. Sci. Comput 2017
- 3D Seismic:
  $m = 1.36$ (FR), $m = 1.19$ (BLR)
  Amestoy et al. SIAM J. Sci. Comput 2015

emgs

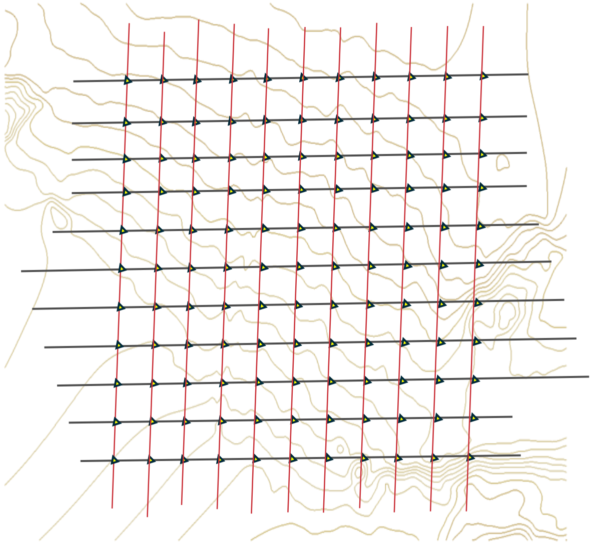
Conclusions

Spot the difference.

Number of RHS estimates

Example CSEM survey over the SEAM model

- $N_r = 11 \times 11 = 121$ receiver
- 22 towlines
- each towline has 150 shot points (30 km / 200 m)
- Source positions in total: $N_s = 22 \times 150 = 3300$
- Field components: $N_{fields} = 4$ (Ex, Ey, Hx, Hy)



BFGS inversion:

$$N_{RHS} = N_r \times N_{fields} \times 2 = 986$$

Gauss-Newton inversion:

$$N_{RHS} = N_s + N_r \times N_{fields} = 3784$$

MUMPS-BLR vs Iterative solver

Inversion	Number of RHS	FR solver times (sec)				BLR solver times (sec)				Iterative solver
		Analysis	Factoriz	Solution	Total	Analysis	Factoriz	Solution	Total	Total
BFGS	968	87	2803	965	3856	103	1113	965	2181	803
Gauss-Newton	3784	87	2803	3772	6663	103	1113	3772	4988	3141

- Iterative solver always wins for BFGS inversion (<1000 RHSs)
- Iterative solver always wins for full-rank MUMPS
- Gauss-Newton: due to BLR factorization became faster than iterative solver! ☺
- MUMPS solution time (1 sec / RHS) is currently slower than iterative solver ☹
- BLR can also be applied to the solution phase, and MUMPS may win at the end ☺

Thanks to the MUMPS team!!

emgs

SPOT THE DIFFERENCE.

Thank you