Efficient Substructured Domain-Decomposition in Inverse Problems using Krylov Subspace Recycling

Boris Martin, Tim Gabriel, Christophe Geuzaine

Université de Liège

August 29, 2023

LA LIBERTÉ DE CHERCHER

Full Waveform Inversion (in the frequency domain)

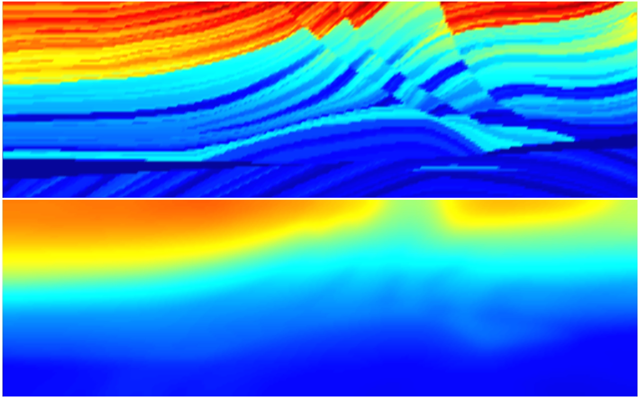


Figure: Slowness squared of the Marmousi model, a common geophysics benchmark for FWI (Target above, initial guess below)

Full Waveform Inversion (in the frequency domain)

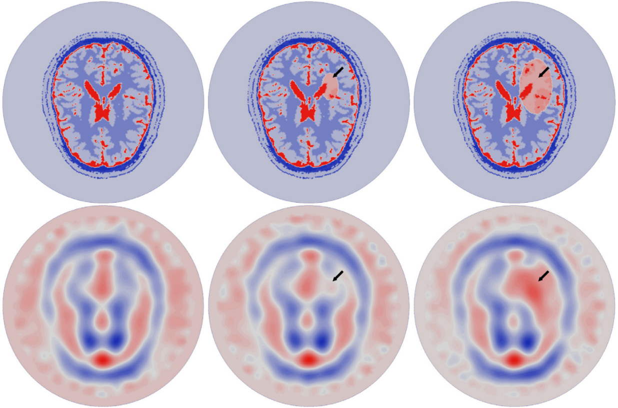


Figure: Imaginary part of permittivity in the brain. *Tournier et al., 2019, Microwave tomographic imaging of cerebrovascular accidents by using high-performance computing*

Full Waveform Inversion (in the frequency domain)

Problem statement: For a model $m(x)$, a wavefield $u(x)$, data d, excitation f and a measurment operator R, find m that minimizes $J(m) = \|Ru(m) - d\|_2^2$ under constraint $A(m)u = f$, where $A(m)$ where A represents the wave equation.

Main cost: solve $A(m)u = f$ for different f and m.

Computing $J(m)$ and $\nabla J(m)$ requires solving 2 wave propagation problems. Then, local optimization (l-BFGS, Newton) is performed.

Domain Decomposition Methods

At large-scale, **direct solvers don't scale** and iterative methods are necessary.
With Domain Decomposition Methods (DDM) we can either:

- Build a preconditioner made of local solves.
- Solve an **interface problem** to glue local solutions together.

Efficient handling of all sources ? Of related models ?

LIÈGE université

EMF 2023

Substructured DDM

Let g be the vector of interface fields and b the physical sources. The *Additive Schwarz Method* leads to a fixed-point scheme.

Propagated incoming wave

$$g^{n+1} = \tilde{A}g^{n} + b$$

New outgoing wave

Local excitation

Instead, solve with **a Krylov solver** (e.g. GMRES, GCR) $Ag = b$ with $A = I - \tilde{A}$.

Substructured DDM

Some properties of the interface problem:

- Significantly lower number of unknowns than the volume problem
- Eigenvalues are in the unit ball centered on 1.
- One matrix-vector product involves solving each subproblem once

Solving all subproblems is the most computationally expensive part.

Efficient FWI

Possible optimizations:

- Handle multiple sources (10, 100, 1000 ?)
- Recycle information when the model changes ?

Example problem

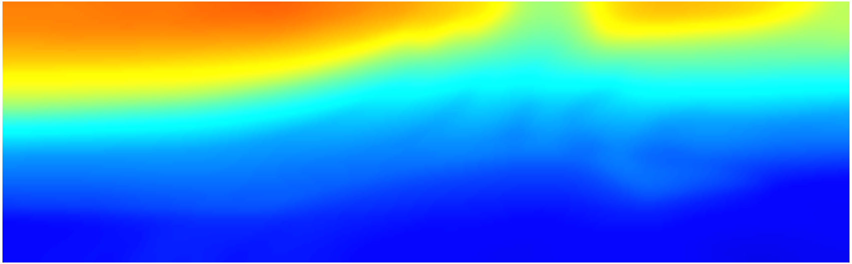
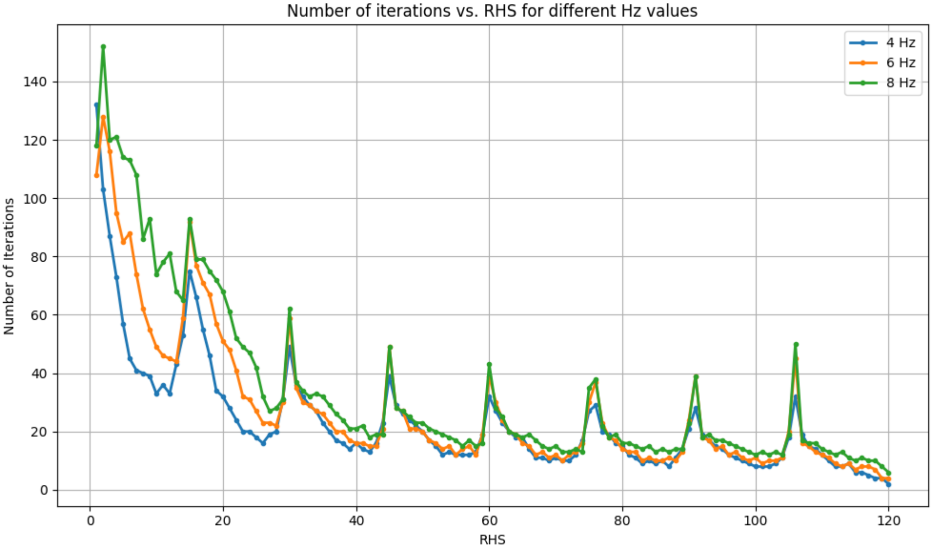


Figure: Test case: 120 sources close to the top. 8x3 subdomains.

Sequential Subspace Recycling

1. Solve $Ag_1 = b_1$ with a solver like GCR.
   - Build an A^*A orthonormal basis of $\mathcal{K}(A, b_1)$: $AU = C, C^*C = I$.
   - Approximation is $g_1 = UC^*b_1$.
2. Reuse this basis **and** expand it to solve $Ag_2 = b_2$.
3. Carry on and accumulate all these search directions.

Results


Number of iterations vs. RHS for different Hz values
4 Hz
6 Hz
8 Hz
Number of Iterations
RHS
0
20
40
60
80
100
120
140
0
20
40
60
80
100
120

Block Krylov Methods

- Solve everything at once, and use the subspace of each RHS in all resolutions.
- Expensive in memory...
- ... But the substructuring makes this bearable!

Influence of block size

Block size	1	10	20	30	60	120
4 Hz	15 580	10 120	7940	6990	5460	3480
6 Hz	12 886	9 930	7700	6810	5580	3600
8 Hz	13 871	11 250	8760	7560	6060	3840

Table: Number of local solves to solve 120 sources on the Marmousi initial model

Results

Table: Number of local solves to solve 120 sources on the Marmousi initial model

	Reference	Recycling	Block	Block deflated
4 Hz	15580	2785	3480	2991
6 Hz	12886	3337	3600	3032
8 Hz	13871	4039	3840	3118

Varying operator

Output of GCR after k steps: directions $U, C \in \mathbb{C}^{n\times k}$ with $AU = C, C^*C = I$.
$\rightarrow UC^*$ is a rank-k approximation of A^{-1}
How to reuse data for a new model ? **Use this approximate inverse as preconditioner** ($+$ correction to make it non-singular)

$$M^{-1} = (I - CC^*) + UC^* = I + (U - C)C^*.$$

Varying operator: Results

Table: Number of local solves to solve 120 sources on 4 similar models.

	Reference	Recycling only	Preconditioner (from 1st)
4 Hz	57 956	12 475	9 496
6 Hz	55 415	16 363	13 759
8 Hz	60 969	20 630	17 440

Truncated Newton methods

Inexact Newton: minimize $J(m)$ by solving with Conjugate Gradients :

$$H\Delta m = -\nabla J.$$

H is the Hessian, and computing Hv for a given v requires 2 additional solves.
The operator is constant but right hand sides are not all available at once
$\rightarrow$More work on the same operator $=$ better recycling

Truncated Newton methods

Perturbed Forward Problem: for a given δA and previously computed u, find δu such that

$$A\delta u = -\delta A u.$$

It's the derivative of the the wavefield with respect to a perturbation.

Truncated Newton methods - Results

For a given model and 5 perturbations: 6 sequences of 120 RHS.

Table: Number of local solves to solve 120 sources and 120×5 perturbations on the Marmousi reconstructed model (3rd iteration)

	Reference	Recycling	Block of 120
4 Hz	94 519	4 246	20 280
6 Hz	78 787	5 778	21 840
8 Hz	87 566	7 362	25 080

Conclusion and future work

- Substructured DDM allows for **greedy subspace recycling**, yielding **fast** convergence.
- Newton methods *could* become more competitive than l-BFGS in this context.

Future work:

- Efficient implementation.
- Further research on changing operator.
- Extension to 3D, more complex equations (Maxwell!).
- Comparison with "DDM as a preconditioner" methods.

References

1. B. Thierry, A. Vion, S. Tournier, M. El Bouajaji, D. Colignon, N. Marsic, X. Antoine, C. Geuzaine. *GetDDM: An open framework for testing optimized Schwarz methods for time-harmonic wave problems*. Computer Physics Communications, Vol. 203, 309-330, 2016. DOI: 10.1016/j.cpc.2016.02.030
2. François-Xavier Roux and A. Barka. *Block Krylov Recycling Algorithms for FETI-2LM Applied to Three-Dimensional Electromagnetic Wave Scattering and Radiation*. IEEE Transactions on Antennas and Propagation, Vol. PP, 1-1, Feb 2017. DOI: 10.1109/TAP.2017.2670541
3. Pierre Jolivet and Pierre-Henri Tournier. *Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers*. In: SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 190-203, 2016. DOI: 10.1109/SC.2016.16