Efficient Handling of Multiple Sources in Domain Decomposition Methods for Full Waveform Inversion

Boris Martin and Christophe Geuzaine

Université de Liège

SIAM LA 24, Sorbonne Université, May 13th 2024

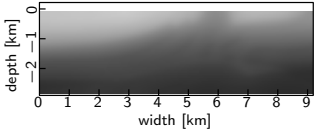
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**

Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**

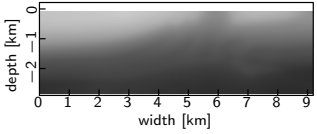
Choose an image



Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**

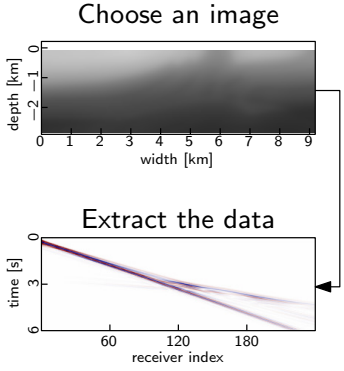
Choose an image



Simulate the propagation

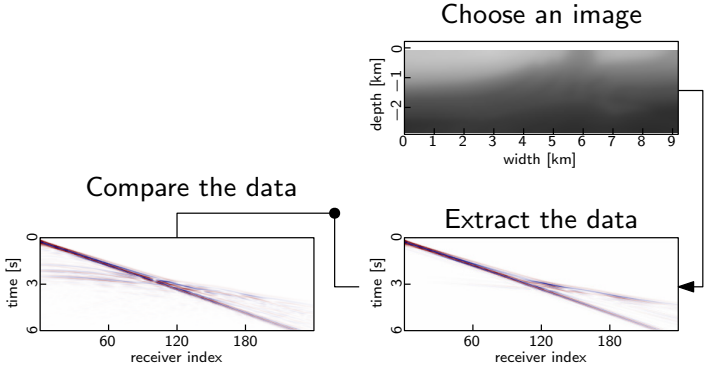
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



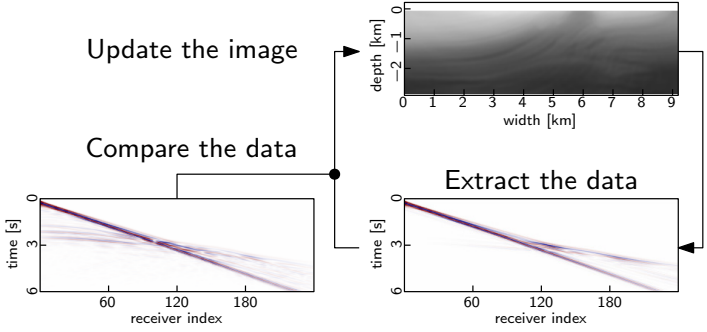
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



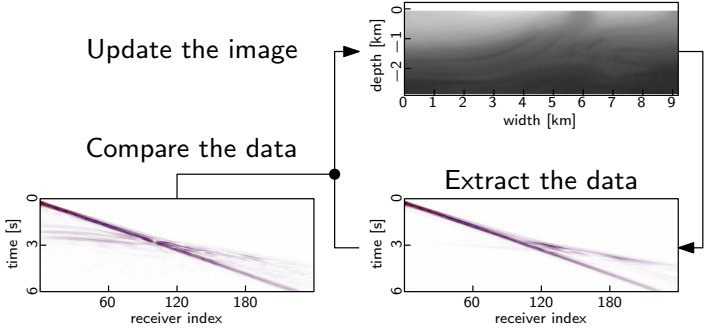
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



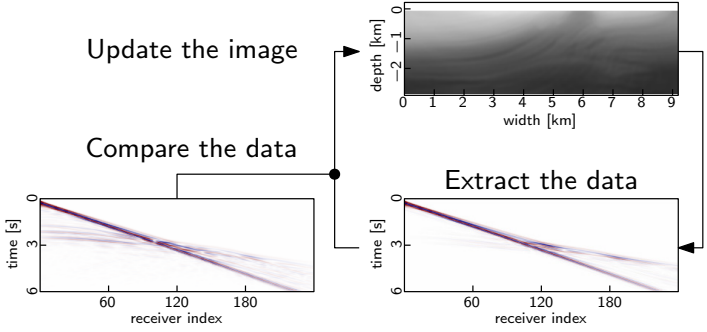
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



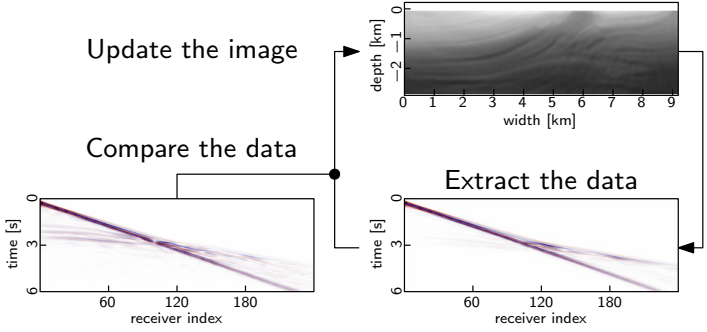
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



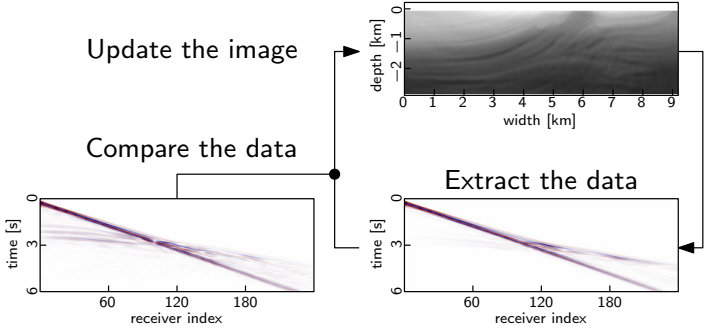
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



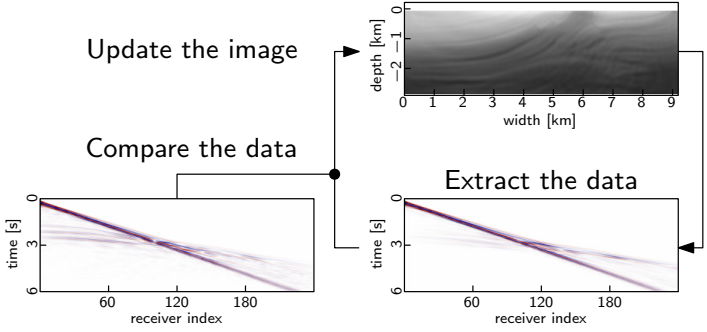
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



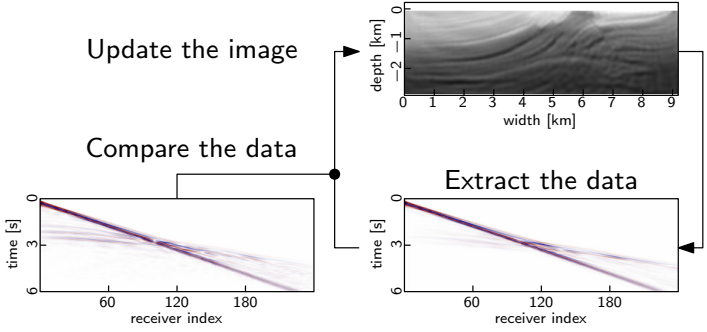
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



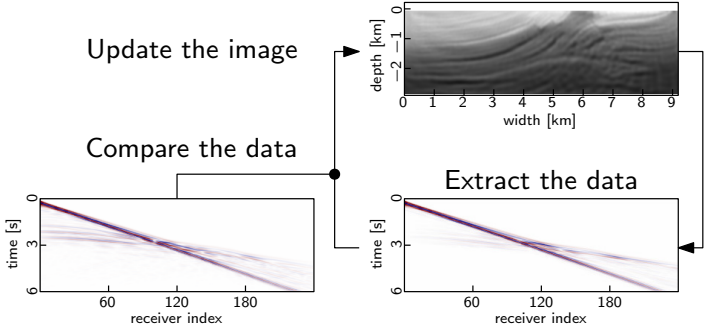
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



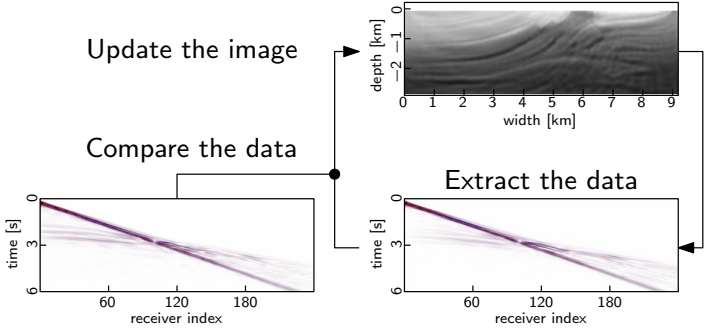
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



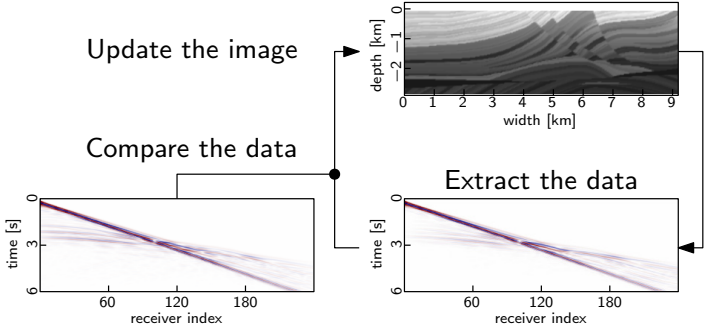
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



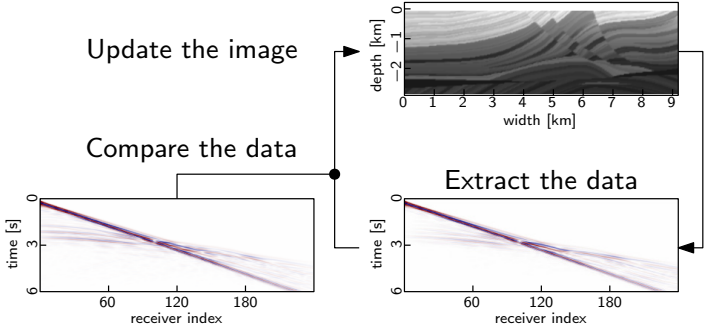
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



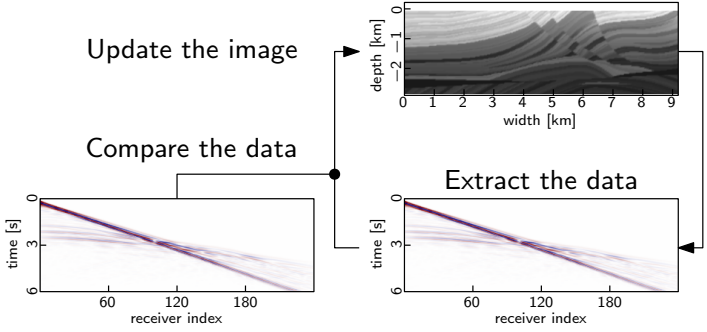
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



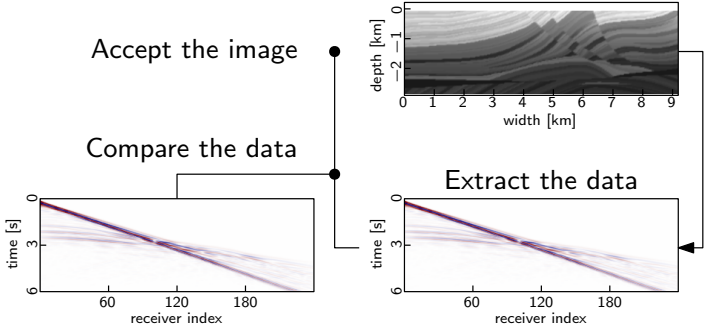
Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



Full waveform inversion (FWI)

FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



[*Adriaens et al. 2023*]

This is FWI in the time domain: we will use it in the **frequency domain**, solving the Helmholtz equation instead of the wave equation

FWI in the frequency domain

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

FWI in the frequency domain

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

Setup for this talk:

- the model $m(x)$ is the local wave speed $c(x)$ in a 2D rectangular domain Ω
- $A(m)$ is the Helmholtz operator, i.e. u satisfies the Helmholtz equation $-\Delta u - \frac{\omega^2}{c(x)^2} u = f$, with ω the angular frequency
- the excitation f consists in (potentially many) point sources located near the top of Ω

FWI in the frequency domain

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

Setup for this talk:

- the model $m(x)$ is the local wave speed $c(x)$ in a 2D rectangular domain Ω
- $A(m)$ is the Helmholtz operator, i.e. u satisfies the Helmholtz equation $-\Delta u - \frac{\omega^2}{c(x)^2} u = f$, with ω the angular frequency
- the excitation f consists in (potentially many) point sources located near the top of Ω

The minimization is carried out using a local, gradient-based optimization method (typically l-BFGS): computing $J(m)$ and $\nabla J(m)$ requires solving 2 Helmholtz problems, using an adjoint approach

FWI in the frequency domain

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

Setup for this talk:

- the model $m(x)$ is the local wave speed $c(x)$ in a 2D rectangular domain Ω
- $A(m)$ is the Helmholtz operator, i.e. u satisfies the Helmholtz equation $-\Delta u - \frac{\omega^2}{c(x)^2} u = f$, with ω the angular frequency
- the excitation f consists in (potentially many) point sources located near the top of Ω

The minimization is carried out using a local, gradient-based optimization method (typically l-BFGS): computing $J(m)$ and $\nabla J(m)$ requires solving 2 Helmholtz problems, using an adjoint approach

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

Domain Decomposition Methods

High-resolution FWI requires $\omega \gg$, leading to large-scale complex and indefinite linear systems for which **direct solvers don't scale** and **standard iterative methods fail** [*Ernst, Gander 2011*]

Domain Decomposition Methods

High-resolution FWI requires $\omega \gg$, leading to large-scale complex and indefinite linear systems for which **direct solvers don't scale** and **standard iterative methods fail** [*Ernst, Gander 2011*]

With Domain Decomposition Methods (DDM) we can either:

- Build a preconditioner made of local solves (e.g. ORAS)
- Solve an **interface problem** to glue local solutions together (e.g. OSM)

Which one to use for FWI?

Model problem: Helmholtz equation

We focus on the Helmholtz equation, but generalizations to other PDEs (Maxwell, elastodynamics) are natural. Using Fourier transform with $e^{-\imath\omega t}$ time dependence, we get:

Helmholtz equation

$$\begin{cases} -\Delta u - k^2 u = f \text{ in } \Omega_i, & \text{(Helmholtz equation)} \\ (\partial_{\boldsymbol{n}} u - \imath k u) = 0, \text{ on } \Gamma^\infty & \text{(radiation condition)} \end{cases}$$

with $k = \frac{\omega}{c(x)}$ the wave number, complex-valued in presence of damping.

Non-Overlapping Optimized Schwarz Method (OSM) for Helmholtz

Partition Ω into **non-overlapping** subdomains Ω_i, $i = 1, \ldots, N_{\text{dom}}$, with interface $\Sigma_{i,j}$ between Ω_i and Ω_j. In each subdomain Ω_i, solve the boundary value problem

Non-overlapping optimized Schwarz formulation

$$\begin{cases} -\Delta u_i - k^2 u_i = f \text{ in } \Omega_i, & \text{(Helmholtz equation)} \\ (\partial_{\mathbf{n}_i} u_i - \imath k u_i) = 0, \text{ on } \Gamma_i^\infty & \text{(radiation condition)} \\ (\partial_{\mathbf{n}_i} u_i - \mathcal{S} u_i) = g_{ij}, \text{ on } \Sigma_{ij} & \text{(interface condition)} \end{cases}$$

with $k = \frac{\omega}{c(x)}$ the wave number and $\mathcal{S}$ a well-chosen interface operator, e.g. $\mathcal{S} = \imath k$. In practice, we use an 2nd order optimized condition:

$$\mathcal{S} = \imath k(\cos\phi + \frac{1}{2k^2} e^{-\imath\phi} \Delta_\Sigma).$$

Experiments show $\phi = \frac{\pi}{2}$ is a good choice.

Non-Overlapping Optimized Schwarz Method (OSM) for Helmholtz

Partition Ω into **non-overlapping** subdomains Ω_i, $i = 1, \ldots, N_{\text{dom}}$, with interface $\Sigma_{i,j}$ between Ω_i and Ω_j. In each subdomain Ω_i, solve the boundary value problem

Non-overlapping optimized Schwarz formulation

$$\begin{cases} -\Delta u_i - k^2 u_i = f \text{ in } \Omega_i, & \text{(Helmholtz equation)} \\ (\partial_{\mathbf{n}_i} u_i - \imath k u_i) = 0, \text{ on } \Gamma_i^\infty & \text{(radiation condition)} \\ (\partial_{\mathbf{n}_i} u_i - \mathcal{S} u_i) = g_{ij}, \text{ on } \Sigma_{ij} & \text{(interface condition)} \end{cases}$$

with $k = \frac{\omega}{c(x)}$ the wave number and $\mathcal{S}$ a well-chosen interface operator, e.g. $\mathcal{S} = \imath k$. In practice, we use an 2nd order optimized condition:

$$\mathcal{S} = \imath k(\cos\phi + \frac{1}{2k^2} e^{-\imath\phi} \Delta_\Sigma).$$

Experiments show $\phi = \frac{\pi}{2}$ is a good choice.

Introduce the interface coupling on Σ_{ij}

$$g_{ij} = -g_{ji} + 2\mathcal{S} u_j := \mathcal{T}_{ji} g_{ji} + b_{ji}$$

Substructured DDM

Rewrite the coupling as a linear system for $g = (g_{ij}, g_{ji})^T$:

$$\underbrace{A}_{\text{iteration matrix}} \quad \underbrace{g}_{\text{interface unknowns}} = \underbrace{b}_{\text{physical sources}}, \quad A = I - \begin{pmatrix} 0 & \mathcal{T}_{ji} \\ \mathcal{T}_{ij} & 0 \end{pmatrix}$$

We solve this linear system with a matrix-free **Krylov solver** such as GMRES or GCR

Substructured DDM

Rewrite the coupling as a linear system for $g = (g_{ij}, g_{ji})^T$:

$$\underbrace{A}_{\text{iteration matrix}} \quad \underbrace{g}_{\text{interface unknowns}} = \underbrace{b}_{\text{physical sources}}, \quad A = I - \begin{pmatrix} 0 & \mathcal{T}_{ji} \\ \mathcal{T}_{ij} & 0 \end{pmatrix}$$

We solve this linear system with a matrix-free **Krylov solver** such as GMRES or GCR

Properties of the interface problem:

- Significantly smaller number of unknowns than the volume problem
- Eigenvalues are in the unit ball centered on 1 for "good" $\mathcal{S}$
- One matrix-vector product involves solving each subproblem once

Solving the subproblems using a sparse direct solver is the most computationally expensive part

Optimized Restrictive Additive Schwarz (ORAS) for Helmholtz

Partition Ω into **overlapping** subdomains Ω_i, $i = 1, \ldots, N_{\text{dom}}$, define a restriction map R_i from Ω to Ω_i and a partition of unity D_i such that $\sum_i^{N_{\text{dom}}} R_i^T D_i R_i = I$. Let A be the Helmholtz operator for a given discretization.

RAS and ORAS preconditioners

Let $A_{\text{loc},i} = R_i A R_i^T$ the local matrix in domain i. The RAS preconditioner is given by

$$M_{\text{RAS}}^{-1} = \sum_{i=1}^{N_{\text{dom}}} R_i^T D_i A_{\text{loc},i}^{-1} R_i.$$

If we replace $A_{\text{loc},i}$ by a local Helmholtz operator $A_{\text{imp},i}$ with impedance (Robin) BCs on the artificial boundaries, we get the improved ORAS preconditioner:

$$M_{\text{ORAS}}^{-1} = \sum_{i=1}^{N_{\text{dom}}} R_i^T D_i A_{\text{imp},i}^{-1} R_i.$$

Optimized Restrictive Additive Schwarz (ORAS) for Helmholtz

Partition Ω into **overlapping** subdomains Ω_i, $i = 1, \ldots, N_{\text{dom}}$, define a restriction map R_i from Ω to Ω_i and a partition of unity D_i such that $\sum_i^{N_{\text{dom}}} R_i^T D_i R_i = I$. Let A be the Helmholtz operator for a given discretization.

RAS and ORAS preconditioners

Let $A_{\text{loc},i} = R_i A R_i^T$ the local matrix in domain i. The RAS preconditioner is given by

$$M_{\text{RAS}}^{-1} = \sum_{i=1}^{N_{\text{dom}}} R_i^T D_i A_{\text{loc},i}^{-1} R_i.$$

If we replace $A_{\text{loc},i}$ by a local Helmholtz operator $A_{\text{imp},i}$ with impedance (Robin) BCs on the artificial boundaries, we get the improved ORAS preconditioner:

$$M_{\text{ORAS}}^{-1} = \sum_{i=1}^{N_{\text{dom}}} R_i^T D_i A_{\text{imp},i}^{-1} R_i.$$

Then, we solve with GMRES (or related) the linear system $M_{\text{ORAS}}^{-1} A u = M_{\text{ORAS}}^{-1} f$.

Comparison of the methods

Substructured DDM (OSM)

- Non-overlapping subdomains
- Solve (with GMRES) a linear system for the interface unknowns (**which is smaller!**)
- Local solves are part of the operator application

ORAS

- Overlapping subdomains
- Invert the global FEM matrix with GMRES
- Use local solves to build a preconditioner

N.B.: The OSM method is also known as FETI-2LM.

Comparison of the methods

Substructured DDM (OSM)

- Non-overlapping subdomains
- Solve (with GMRES) a linear system for the interface unknowns (**which is smaller!**)
- Local solves are part of the operator application

ORAS

- Overlapping subdomains
- Invert the global FEM matrix with GMRES
- Use local solves to build a preconditioner

N.B.: The OSM method is also known as FETI-2LM.

Which one to use for FWI? How to handle multiple sources?

HPC considerations: cost of GMRES

To solve $M^{-1}Ax = M^{-1}b$ with GMRES, at iteration j, the main costs are:

- Do one matix-vector product with A.
- Apply M^{-1} once.
- Do j inner products to build an orthonormal basis of the search space.

HPC considerations: cost of GMRES

To solve $M^{-1}Ax = M^{-1}b$ with GMRES, at iteration j, the main costs are:

- Do one matix-vector product with A.
- Apply M^{-1} once.
- Do j inner products to build an orthonormal basis of the search space.

The relative cost of each part depends on the the operators, the problem size and the iteration count.

- The orthogonalization cost is higher in ORAS than in OSM, because the search space is larger.
- Subdomain solves are more expensive in ORAS because of the overlap.
- Overlap should help convergence.

HPC considerations: cost of GMRES for multiple RHS

When many sources are involved, the matrix assembly and factorization costs are amortized over many right-hand sides. The additional cost per source is then only the cost of the GMRES iterations.

Is applying naively GMRES to each source the best strategy?

Block GMRES for multiple right hand sides

Let p be the number of sources and m a number of iterations.
For each source b_p, we build an orthonormal basis of $K^m(M^{-1}A, b_p)$, whose cost is $(O)m^2$.
The total cost is thus $\mathcal{O}(pm^2)$.
Then, x_p is sought in the p-th space. Why not share information between the sources?

GMRES allows us to search for $x_1, x_2, \ldots, x_p$ in the direct sum of the p subspaces, a large space of size pm.

- The cost of the orthogonalization is $\mathcal{O}(p^2m^2)$.
- The approximation at iteration m is the best approximation in the pm-dimensional space.
- How much the convergence improves is problem-dependent.

Can BGMRES reduce the iteration count enough to mitigate the higher orthogonalization cost?

Research questions summarized

- Is the convergence impact of blocks similar with ORAS and OSM?
- Are blocks cheaper with non-overlapping methods?
- What strategy among the four is the best for our problems?

Research questions summarized

- Is the convergence impact of blocks similar with ORAS and OSM?
- Are blocks cheaper with non-overlapping methods?
- What strategy among the four is the best for our problems?

Our hypothesis at first:

- Convergence impact should be similar in both cases. (Similar physical intuition)
- Block should be significantly cheaper with non-overlapping methods.
- Without blocks, we expect ORAS to be faster, but with blocks and enough sources, OSM could become faster.

Effect of Substructuring

How much smaller is the reduced problem ?
It depends on the surface/volume ratio of the subdomains. Typically, it will be 5 to 10 times smaller (for subdomains of aroud 100k DOFs).

Model problem

We try to solve the Helmholtz equation in a 3D box (13.5 km $\times$ 13.5 km $\times$ 4.18 km), with 64 sources on top of it. (8 $\times$ 8 array on a plane)
We use:

- Either ORAS or OSM.
- Either GMRES or BGMRES.
- A cartesian partition of the domain into boxes.
- Tetrahedral elements with P2 or P3 basis functions.

Cost of orthogonalization and blocks: example case

Test case: Homogeneous velocity of 2 km/s, and 1 Hz as frequency. (Wavelength of 2 km) Mesh size of 130 m, second order polynomial basis on tetraheda, 64 subdomains (4 in each direction), with a total of 3.1 million DOFs.

Cost of orthogonalization and blocks: example case

Test case: Homogeneous velocity of 2 km/s, and 1 Hz as frequency. (Wavelength of 2 km) Mesh size of 130 m, second order polynomial basis on tetraheda, 64 subdomains (4 in each direction), with a total of 3.1 million DOFs.

Hardware: One node of Lemaitre4 (CÉCI cluster at UC Louvain), 2x AMD EPYC 9534 64-Core CPUs, 766 GB of RAM. We use 64 processes with 2 threads.

Software: Gmsh + GmshFEM + GmshDDM + PETSc with HPDDM (for multi-RHS GMRES and BGMRES).

Cost of orthogonalization and blocks: example case

We first perform 30 iterations without looking at the convergence. We only measure the cost of subdomain solves and the Krylov method.

	GMRES-ORAS	BGMRES-ORAS	GMRES-OSM	BGMRES-OSM
Factorization	8s		6s	
Local Solves	43s		24s	
Orthogonalization	25s	43s	6s	10s
DOFs per process	$\approx$ 80k		$\approx$ 50k (Volume), 10k (interface)	
Global size	3.1×10^6		6×10^5	
Combined time	76s	94s	36s	40s

Cost of orthogonalization and blocks: example case

We first perform 30 iterations without looking at the convergence. We only measure the cost of subdomain solves and the Krylov method.

	GMRES-ORAS	BGMRES-ORAS	GMRES-OSM	BGMRES-OSM
Factorization	8s		6s	
Local Solves	43s		24s	
Orthogonalization	25s	43s	6s	10s
DOFs per process	$\approx$ 80k		$\approx$ 50k (Volume), 10k (interface)	
Global size	3.1×10^6		6×10^5	
Combined time	76s	94s	36s	40s

- Blocks are **not** 64 times more expensive in the orthogonalization (higher arithmetic intensity etc.)
- Substructured methods are 5 times faster in the orthogonalization, because the problem is 5 times smaller.
- Optimized assembly needed of OSM needed in the future.

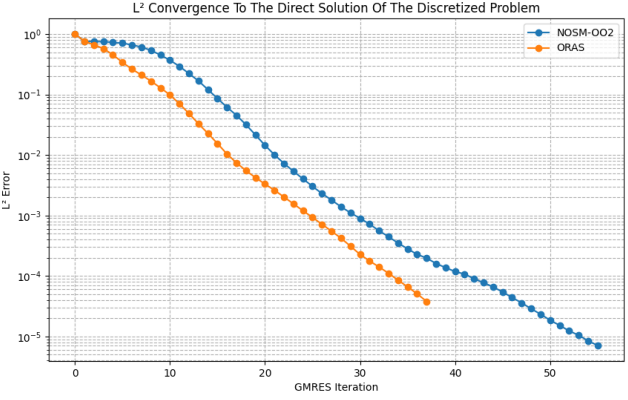
Intermediate conclusions

- ✓ Substructuring efficiently reduces the cost of orthogonalization.
- ✓ Block methods are affordable (in time) on real hardware[1], especially with substructured methods.
- Impact on convergence?
- Global conclusion on what approach to use?
- Not taken into account: lower memory cost of non-overlapping methods.

[1] With a Cholesky QR, BGMRES doesn't require more synchronizations than GMRES

Convergence of ORAS and OSM for 1 RHS

ORAS and OSM minimize different residuals: what metric to use for comparing accuracy? **We use the L^2 error on the solution from a direct solver.** We set a tolerance of 10^{-6} for the ORAS preconditioned residual and 10^{-4} for the OSM residual.



L^2-convergence of ORAS and OSM for a homogeneous medium and the prescribed tolerances.

Convergence of ORAS and OSM for 64 RHSs

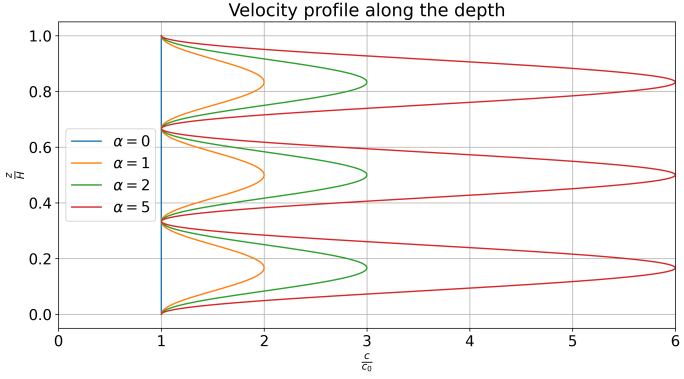
How fast can we solve the 64 RHSs with and without block?
The cost of blocks depends on the problem size and partitioning, **but the convergence impact depends heavily on the problem.** Let's try with a medium with a paramatric heterogeneity.

Effectiveness on heterogeneous media

Toy problem: oscillating velocity along the z-axis ($c_0 = 2000$ m/s).

$$c(x, y, z) = c_0 \left(1 + \alpha \sin^2(3\pi \frac{z}{H})\right), \ 0 < z < H.$$

We compare the 4 approaches for various values of α, including 0 (homogeneous).



Results for a low frequency

Iteration count (time in s) for $f = 1$ Hz, mesh size of 200 m and P2 elements. 1.1 million DOFs, 284 000 on the interface problem.

α	ORAS-GMRES	ORAS-BGMRES	OSM-GMRES	OSM-BGMRES
0	35 (31)	20 (23)	44 **(16)**	37 (17)
0.5	35 (32)	18 (20)	49 **(18)**	39 **(18)**
1	40 (35)	17 (19)	59 (22)	42 **(19)**
2	47 (40)	17 **(19)**	98 (34)	48 **(21)**
5	66 (57)	16 **(17)**	> 200	59 (25)
10	57 (49)	16 **(18)**	> 300	75 (32)

Results for a low frequency

Iteration count (time in s) for $f = 1$ Hz, mesh size of 200 m and P2 elements. 1.1 million DOFs, 284 000 on the interface problem.

α	ORAS-GMRES	ORAS-BGMRES	OSM-GMRES	OSM-BGMRES
0	35 (31)	20 (23)	44 **(16)**	37 (17)
0.5	35 (32)	18 (20)	49 **(18)**	39 **(18)**
1	40 (35)	17 (19)	59 (22)	42 **(19)**
2	47 (40)	17 **(19)**	98 (34)	48 **(21)**
5	66 (57)	16 **(17)**	> 200	59 (25)
10	57 (49)	16 **(18)**	> 300	75 (32)

- Overlapping methods are more robust to heterogeneities.
- Blocks are particularly effective for high α.

Results for a medium frequency

Iteration count (time in s) for $f = 2\,\text{Hz}$, mesh size of 130 m and P2 elements. 3.1 million DOFs, 602 000 on the interface problem. A direct solve takes 340s.

α	ORAS-GMRES	ORAS-BGMRES	OSM-GMRES	OSM-BGMRES
0	38 (91)	28 (87)	43 (50)	37 **(47)**
0.5	37 (85)	18 (87)	43 **(50)**	39 (51)
1	42 (98)	27 (82)	47 **(52)**	42 (55)
2	45 (98)	23 (71)	55 (59)	44 **(57)**
5	30 (75)	21 (64)	90 (92)	48 **(60)**
10	40 (93)	22 **(70)**	> 500	60 ($>$75)

Results for a medium frequency

Iteration count (time in s) for $f = 2\,\text{Hz}$, mesh size of 130 m and P2 elements. 3.1 million DOFs, 602 000 on the interface problem. A direct solve takes 340s.

α	ORAS-GMRES	ORAS-BGMRES	OSM-GMRES	OSM-BGMRES
0	38 (91)	28 (87)	43 (50)	37 **(47)**
0.5	37 (85)	18 (87)	43 **(50)**	39 (51)
1	42 (98)	27 (82)	47 **(52)**	42 (55)
2	45 (98)	23 (71)	55 (59)	44 **(57)**
5	30 (75)	21 (64)	90 (92)	48 **(60)**
10	40 (93)	22 **(70)**	> 500	60 (>75)

- Under-resolved mesh, probably with some pollution.
- Non-overlapping are more effective on under-resolved meshes.
- Blocks are not worth it for ORAS with small α.

Results for a high frequency

Iteration count (time in s) for $f = 3$ Hz, mesh size of 160 m and P3 elements. 6.7 million DOFs, 1 million on the interface problem.

α	ORAS-GMRES	ORAS-BGMRES	OSM-GMRES	OSM-BGMRES
0	34 (241)	27 (229)	45 **(109)**	41 (111)
0.5	38 (270)	29 (247)	**51 (118)**	45 (121)
1	39 (257)	30 (267)	54 (126)	47 **(125)**
2	45 (307)	28 (240)	65 (150)	51 **(133)**
5	48 (332)	17 (161)	75 (168)	54 **(138)**
10	16 (167)	12 **(124)**	97 (210)	54 (140)

A word on memory consumption

Memory cost of BGMRES or GMRES with p sources and m iterations: pm vectors.
For $p = 64, m = 30$ and a problem of 5 millions DOFs, it's around 150 GB! Substructuring can reduce this by a factor of 5 to 10.

ORAS-BGMRES might be too memory-intensive on real applications.

A word on memory consumption

Memory cost of BGMRES or GMRES with p sources and m iterations: pm vectors.
For $p = 64, m = 30$ and a problem of 5 millions DOFs, it's around 150 GB! Substructuring can reduce this by a factor of 5 to 10.

ORAS-BGMRES might be too memory-intensive on real applications.
Intermediate approach: BGMRES or BGCRODR (supsbace recycling) on smaller blocks.

Conclusion

We compared two domain decomposition methods for solving the Helmholtz equation, ORAS and OSM.

For both approaches, we assessed the relevance of using BGMRES instead of GMRES.

- ✓ Substructuring efficiently reduces the cost of orthogonalization.
- ✓ Block methods sometimes have a significant boost on convergence, especially in highly heterogeneous media.
- × The best method is heavily problem-dependent.
- × Optimized implementation needed for a fair benchmarking.

Future work & open questions

- Combination with subspace recycling (BGCRO-DR, Block Orthodir with rank revealing QR at each iteration).
- Interaction with coarse grids, such as GenEO.
- Unstructured (automatic) partitioning.
- Comparison with the best transmission conditions (PML, Padé).
- Recycling from one velocity model to another.
- Substructured overlapping methods?

Future work & open questions

- Combination with subspace recycling (BGCRO-DR, Block Orthodir with rank revealing QR at each iteration).
- Interaction with coarse grids, such as GenEO.
- Unstructured (automatic) partitioning.
- Comparison with the best transmission conditions (PML, Padé).
- Recycling from one velocity model to another.
- Substructured overlapping methods?

Thanks for your attention

✉ boris.martin@uliege.be

Effect of Substructuring

How much smaller is the reduced problem ?
It depends on the surface/volume ratio of the subdomains. Rule of thumb: for a cube of size n, the surface is $6n^2$ and the volume is n^3.
Surface to volume ratio is then $6/n$.
For domains around 50k to 100k DOFs, **the reduction factor is typically between 5 and 10**.

(In 2D, the reduction factor can go beyond 100!)

Neglected aspects

- For ORAS: Sparse matrix-vector product. (Cheap and efficiently implemented in PETSc)
- OSM: assembly of the interface problem. Very expensive in our implementation but easily optimized: replace the assembly by an explicit sparse matrix-vector product.
- We have not considered variable overlap sizes.
- Optimized transmission conditions for ORAS were not used, for simplicity.