Efficient Handling of Multiple Sources in Non-Overlapping Domain Decomposition Methods for Full Waveform Inversion in the Frequency Domain

Boris Martin, Tim Gabriel and Christophe Geuzaine

Université de Liège

DD28, KAUST, January 31 2024

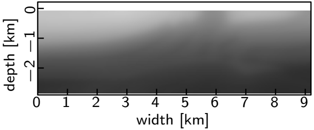
Full waveform inversion (FWI)

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**

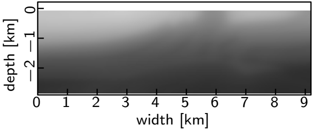
Choose an image



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**

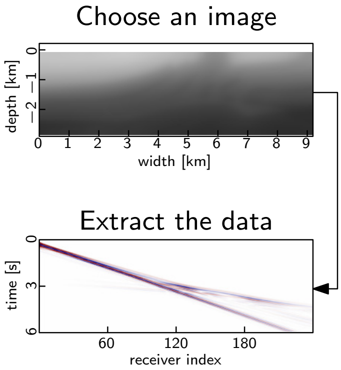
Choose an image



Simulate the propagation

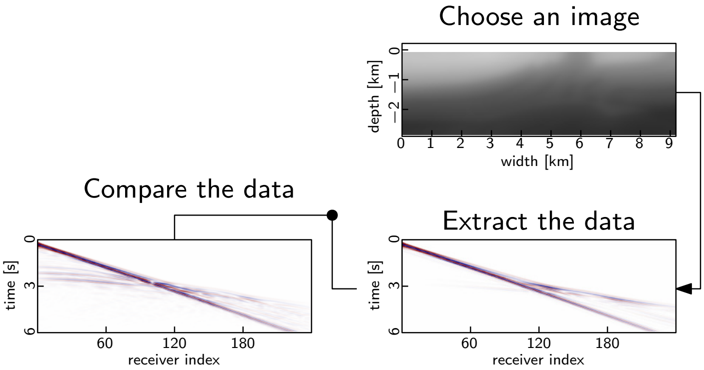
Full waveform inversion (FWI)

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**



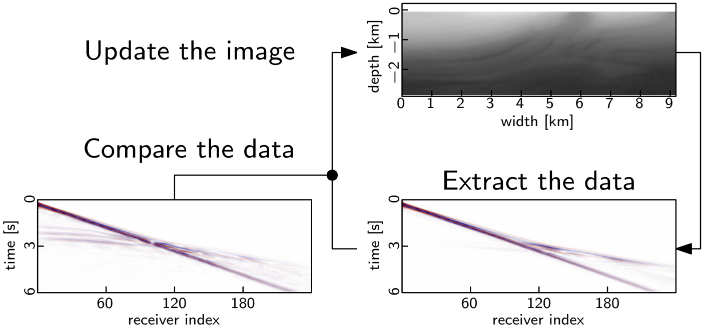
Full waveform inversion (FWI)

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**



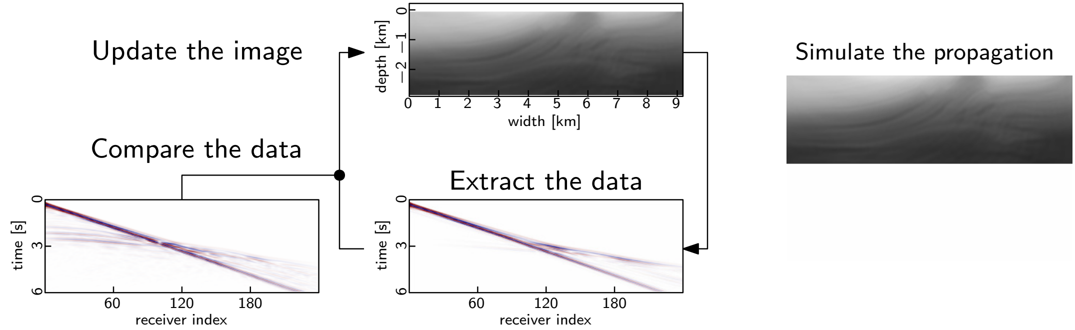
Full waveform inversion (FWI)

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**



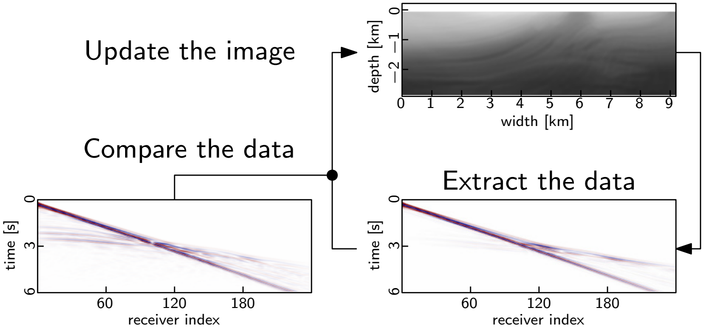
Full waveform inversion (FWI)

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**



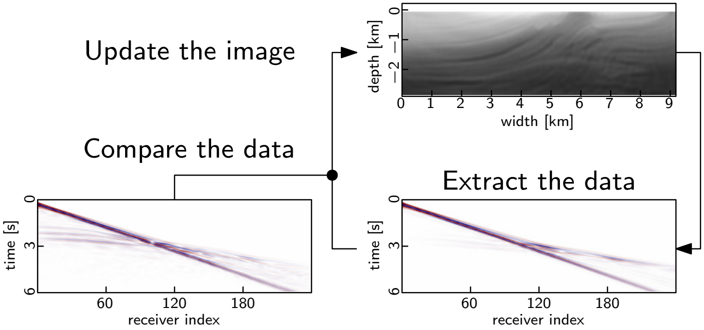
Full waveform inversion (FWI)

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**



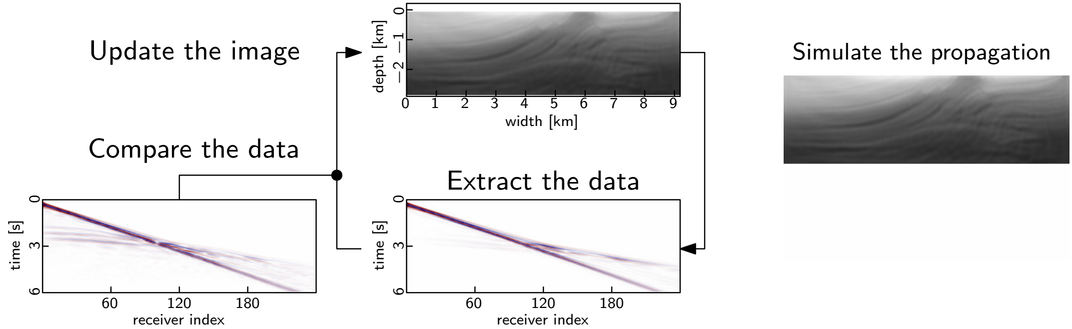
Full waveform inversion (FWI)

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**



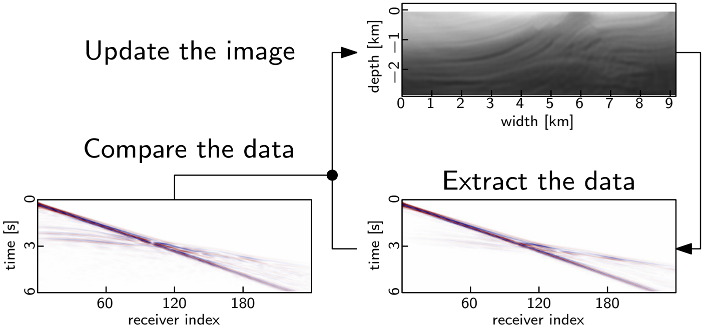
Full waveform inversion (FWI)

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**



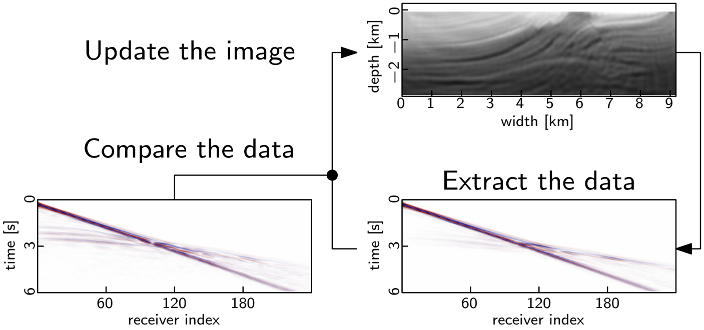
Full waveform inversion (FWI)

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**



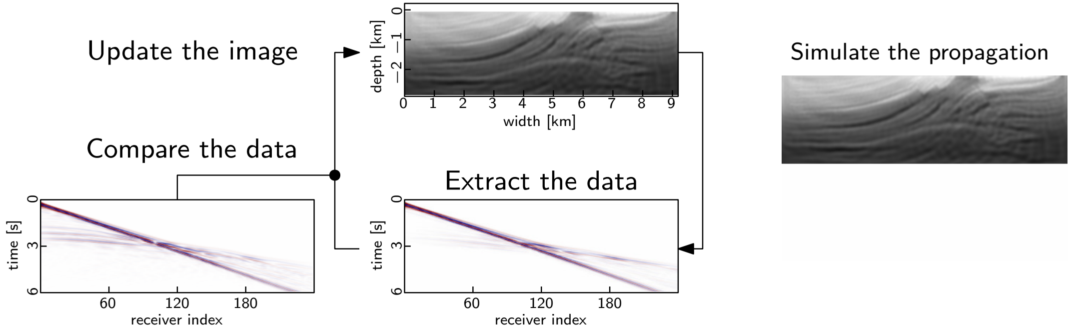
Full waveform inversion (FWI)

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**



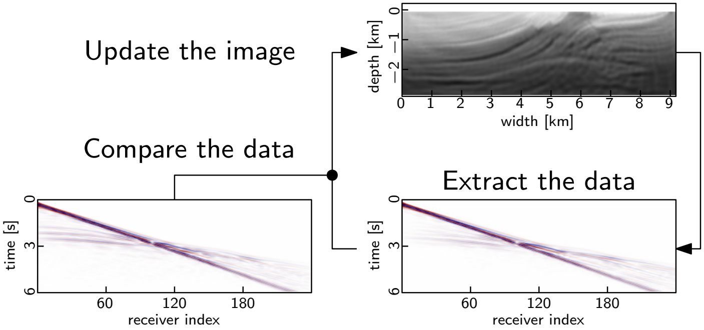
Full waveform inversion (FWI)

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**



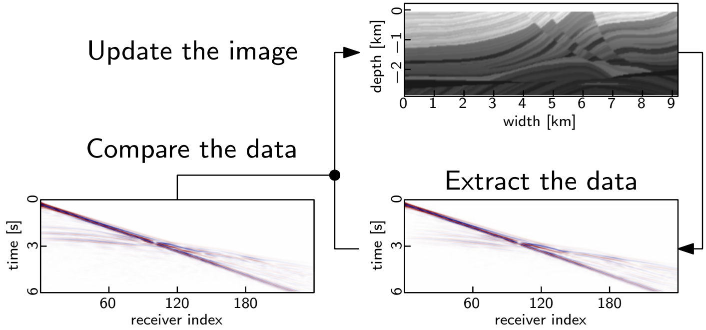
Full waveform inversion (FWI)

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**



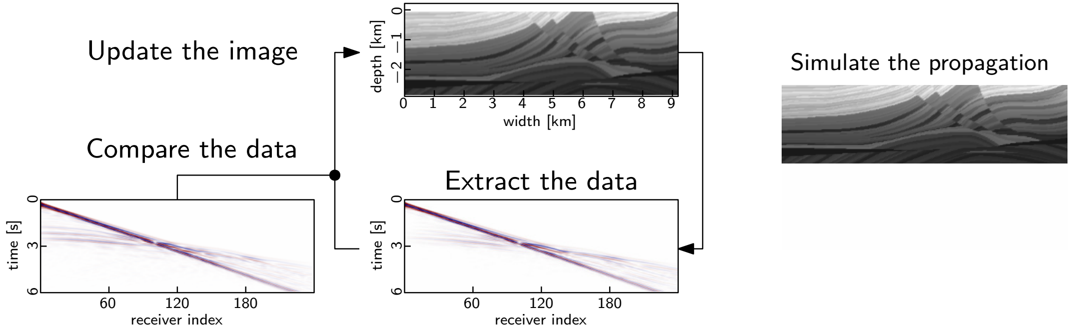
Full waveform inversion (FWI)

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**



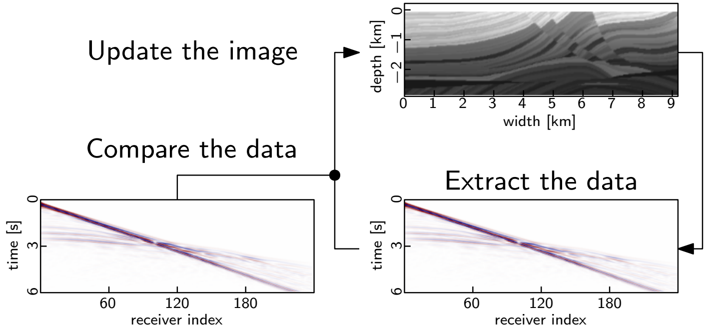
Full waveform inversion (FWI)

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**



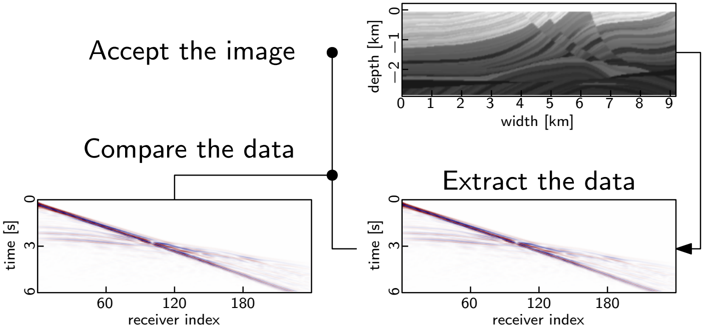
Full waveform inversion (FWI)

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

We focus on the latter

Non-Overlapping Schwarz DDM for Helmholtz

Partition Ω into non-overlapping subdomains Ω_i, $i = 1, \dots, N_{\text{dom}}$, 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 (simplest: $\mathcal{S} = ik$)

Non-Overlapping Schwarz DDM 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 (simplest: $\mathcal{S} = ik$)

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 $g = (g_{ij}, g_{ji})^T$:

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

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 $g = (g_{ij}, g_{ji})^T$:

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

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

Efficient FWI

For an efficient resolution of the inverse problem, how to:

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

Efficient FWI

For an efficient resolution of the inverse problem, how to:

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

Build upon:

- P. Jolivet and P.-H. Tournier. *Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers*. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2016

Efficient FWI

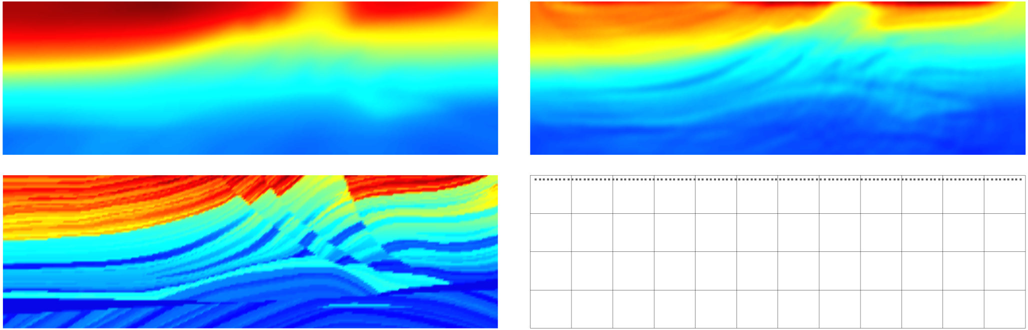
For an efficient resolution of the inverse problem, how to:

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

Build upon:

- P. Jolivet and P.-H. Tournier. *Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers*. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2016
- F.-X. 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, 2017

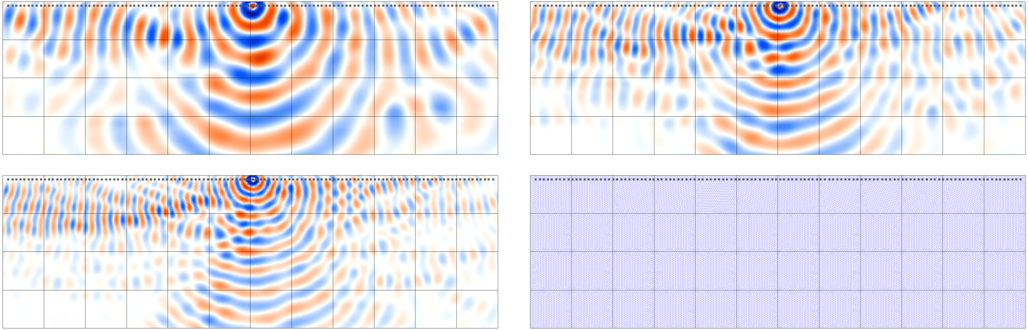
Benchmark problem: Marmousi model



Slowness squared ($\frac{1}{c(x)^2}$): initial model, one FWI iteration, target; DDM partitions

- 120 equidistant sources close to the top
- $48 = 12 \times 4$ subdomains
- Finite element order 2, 3, and 4 at frequencies 4, 6 and 8 Hz, respectively
- Implementation: GmshFEM + GmshDDM + PETSc + HPDDM

Benchmark problem: Marmousi model



Wave fields at 4, 6 and 8 Hz for a single source; finite element mesh

- 120 equidistant sources close to the top
- $48 = 12 \times 4$ subdomains
- Finite element order 2, 3, and 4 at frequencies 4, 6 and 8 Hz, respectively
- Implementation: GmshFEM + GmshDDM + PETSc + HPDDM

Sequential subspace recycling with GCR

The GCR algorithm is a minimal residual Krylov solver (equivalent to GMRES) that builds a A^*A orthonormal basis of a subspace. To solve $Ax = b$:

1. Set $r_0 = b - Ax_0$
2. For $i = 1, 2, \ldots$ until convergence, do:
   - → Pick a new direction $\tilde{u}_i = r_{i-1}$ and set $\tilde{c}_i = A\tilde{u}_i$.
   - → Make it A^*A orthogonal to previous directions u_j $(j < i)$ with a Gram-Schmidt procedure:

$$y_j = c_j^* \tilde{c}_i$$

$$\tilde{u}_i := \tilde{u}_i - \sum_{j<i} u_j y_j$$

   - → Normalize it to get u_i and $c_i = Au_i$ such that $c_i^* c_i = 1$.
   - → Compute step length $\alpha_i = c_i^* r_{i-1}$.
   - → Set $x_i = x_{i-1} + \alpha_i u - i$ and $r_i = r_{i-1} - \alpha_i c_i$.

Sequential subspace recycling with GCR

The procedure extends naturally to sequences of right hand sides

Let U and $C = AU$ contain the columns of the previous directions. To solve for another b:

1. Set $x_0 = UC^*b$
2. For $i = 1, 2, ...$ until convergence, do:
   - $\rightarrow$ Pick a new direction $\tilde{u}_i = r_{i-1}$ and set $\tilde{c}_i = A\tilde{u}_i$.
   - $\rightarrow$ Make it A^*A orthogonal to directions from the previous RHS: $\tilde{u}_i := \tilde{u}_i - UC^*\tilde{u}_i$.
   - $\rightarrow$ Make it A^*A orthogonal to previous directions from this RHS as previously.
   - $\rightarrow$ Normalize, compute step length and update x_i as before.

Sequential subspace recycling with GCR

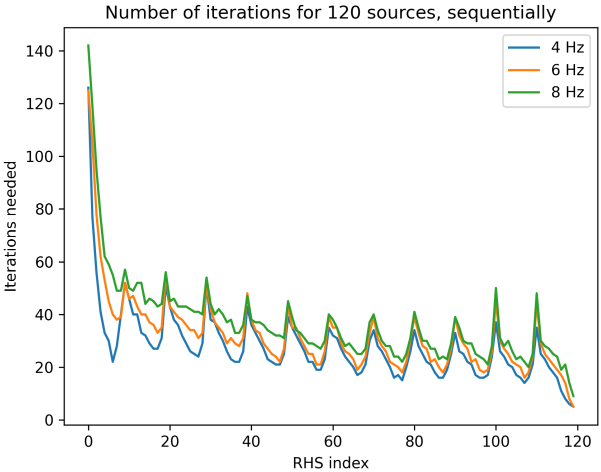
The procedure extends naturally to sequences of right hand sides

Let U and $C = AU$ contain the columns of the previous directions. To solve for another b:

1. Set $x_0 = UC^* b$
2. For $i = 1, 2, \ldots$ until convergence, do:
   - $\rightarrow$ Pick a new direction $\tilde{u}_i = r_{i-1}$ and set $\tilde{c}_i = A\tilde{u}_i$.
   - $\rightarrow$ Make it A^*A orthogonal to directions from the previous RHS: $\tilde{u}_i := \tilde{u}_i - UC^*\tilde{u}_i$.
   - $\rightarrow$ Make it A^*A orthogonal to previous directions from this RHS as previously.
   - $\rightarrow$ Normalize, compute step length and update x_i as before.

For each additional RHS, convergence is (hopefully) **faster** but the space size keeps **increasing**

Sequential subspace recycling with GCR



- Significant gains, even for modest number of sources
- As a comparison, without recycling, the average number of iterations per RHS is: 124 at 4Hz, 122 at 6 Hz, and 138 at 8Hz

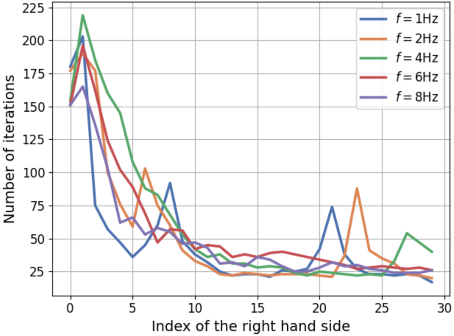
Robustness of sequential recycling

- Increased number of iterations when sources get close to subdomain interfaces
- Increased number of iterations with frequency

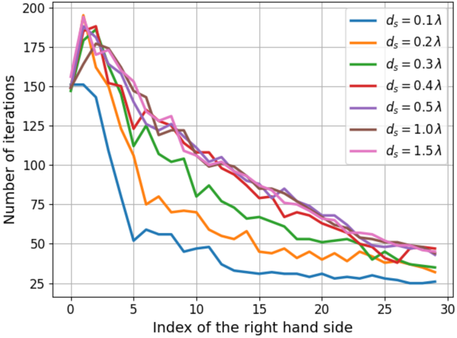
Robustness of sequential recycling

- Increased number of iterations when sources get close to subdomain interfaces

- Increased number of iterations with frequency
  → variation is actually due to distance between sources **relative to the wavelength** λ



Fix $d_s = 0.2\lambda$



Robustness of sequential recycling

- Choice of transmission operator $\mathcal{S}$, mesh refinement and finite element order have only marginal impact
- If one caps the number of directions, the simplest recycling strategy (recycling the first directions) is the best; it outperforms recycling
  - $\rightarrow$ the last (most recent) directions
  - $\rightarrow$ directions leading to the largest residual decrease
  - $\rightarrow$ directions leading to the most significant coefficients in absolute value during the orthogonalization

Block Krylov methods

We can also use Block GMRES (BGMRES) for faster convergence:

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

Block Krylov methods

We can also use Block GMRES (BGMRES) for faster convergence:

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

Number of local solves (matrix-vector product) for solving the same 120 sources on the Marmousi initial model

Block size	1	5	10	30	60	120
4 Hz	15 017	8 950	7 440	6 240	5 460	3 840
6 Hz	14 838	9 205	7 130	5 520	4 800	3 360
8 Hz	16 663	10 650	8 090	5 670	4 800	3 240

Block Krylov methods

We can also use Block GMRES (BGMRES) for faster convergence:

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

Number of local solves (matrix-vector product) for solving the same 120 sources on the Marmousi initial model

Block size	1	5	10	30	60	120
4 Hz	15 017	8 950	7 440	6 240	5 460	3 840
6 Hz	14 838	9 205	7 130	5 520	4 800	3 360
8 Hz	16 663	10 650	8 090	5 670	4 800	3 240

- Larger blocks always lead to a faster convergence
- Robust w.r.t. the choice of interface operator $\mathcal{S}$

Comparison of the two approaches

Number of local solves for solving the 120 sources on the Marmousi initial model

	Reference	Sequential	Full Block
4 Hz	15 017	3 440	3 840
6 Hz	14 838	3 972	3 360
8 Hz	16 663	4 552	3 240

Comparison of the two approaches

Number of local solves for solving the 120 sources on the Marmousi initial model

	Reference	Sequential	Full Block
4 Hz	15 017	3 440	3 840
6 Hz	14 838	3 972	3 360
8 Hz	16 663	4 552	3 240

- Both approaches are useful
  - $\rightarrow$ ... and give comparable speedups
- Recycling seems more sensitive to frequency (number of wavelengths between sources)
- Block Krylov needs all RHS available at once

Is it worth it?

- In compute time: orthogonalization cost is dwarfed by the subdomain solve time
- In memory: for large enough subdomains, memory cost is small compared to the LU storage
- Effectiveness in 3D to be confirmed, but the reasoning is similar
- Stability: large blocks / sequences require an accurate orthogonalization scheme, which can be expensive

What about variations in the operator?

Output of GCR after k steps: directions $U, C \in \mathbb{C}^{n\times k}$, with $AU = C$ and $C^*C = I$

$\rightarrow$ UC^* is a rank-k approximation of A^{-1}

How to reuse data for a new model?

What about variations in the operator?

Output of GCR after k steps: directions $U, C \in \mathbb{C}^{n\times k}$, with $AU = C$ and $C^* C = I$

$\rightarrow$ UC^* is a rank-k approximation of A^{-1}

How to reuse data for a new model?

- Construct an approximate inverse as **preconditioner**
- Regularized by adding the identity on the orthogonal complement of its nullspace

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

Varying operator: preliminary results

Number of local solves for solving the 120 sources in the first 5 FWI iterations (l-BFGS).
Preconditioner built after solving the 120 sources from the first model.

	Ref	GCR	GCR + Prec	BGMRES	BGMRES + Prec
4 Hz	74 850	17 265	**13 608**	21 000	14 280
6 Hz	73 114	20 047	**17 088**	19 680	17 160
8 Hz	81 801	23 153	20 059	20 400	**17 880**

Varying operator: preliminary results

Number of local solves for solving the 120 sources in the first 5 FWI iterations (l-BFGS).
Preconditioner built after solving the 120 sources from the first model.

	Ref	GCR	GCR + Prec	BGMRES	BGMRES + Prec
4 Hz	74 850	17 265	**13 608**	21 000	14 280
6 Hz	73 114	20 047	**17 088**	19 680	17 160
8 Hz	81 801	23 153	20 059	20 400	**17 880**

- Further (modest) gains can be achieved with the preconditioner
- Stability with respect to the perturbation amplitude is still under study

Conclusion and future work

- Substructured DDM in FWI allows for **greedy subspace recycling** and efficient block-Krylov use, yielding significantly **faster** convergence
- Recycling with a different model is non trivial but still beneficial
- Inexact Newton methods *could* benefit even more than l-BFGS in this context (more work on the same operator, but RHS not available all at once favoring sequential recycling)

Conclusion and future work

- Substructured DDM in FWI allows for **greedy subspace recycling** and efficient block-Krylov use, yielding significantly **faster** convergence
- Recycling with a different model is non trivial but still beneficial
- Inexact Newton methods *could* benefit even more than l-BFGS in this context (more work on the same operator, but RHS not available all at once favoring sequential recycling)

Ongoing and future work:

- Explore robustness of preconditionner for changing operator
- Systematic study in 3D and with different physics (electromagnetics, elasticity)
- Compare with “DDM as a preconditioner” methods, such as ORAS
- Investigate the interactions with coarse grids, such as GenEO

Conclusion and future work

- Substructured DDM in FWI allows for **greedy subspace recycling** and efficient block-Krylov use, yielding significantly **faster** convergence
- Recycling with a different model is non trivial but still beneficial
- Inexact Newton methods *could* benefit even more than l-BFGS in this context (more work on the same operator, but RHS not available all at once favoring sequential recycling)

Ongoing and future work:

- Explore robustness of preconditionner for changing operator
- Systematic study in 3D and with different physics (electromagnetics, elasticity)
- Compare with "DDM as a preconditioner" methods, such as ORAS
- Investigate the interactions with coarse grids, such as GenEO

Thanks for your attention

✉ cgeuzaine@uliege.be

Truncated Newton methods

FWI is typically performed with gradient-based algorithms. Truncated Newton methods are also popular but usually slower.
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. Can this make Newton a more competitive optimization algorithm?

Truncated Newton methods

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

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

This is needed to compute the action of the Hessian on that perturbation.
It is the derivative of the the wave field with respect to a perturbation.
(Neglected here: perturbation of the adjoint state)

Truncated Newton methods - Results

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

Number of local solves for solving the 120 sources and 120×5 perturbations on the Marmousi reconstructed model (3rd iteration of FWI)

	Reference	GCR + Recycling	Block of 120
4 Hz	90 221	6 670	23 880
6 Hz	88 391	8 060	22 080
8 Hz	xxx	xxx	33 120