Numerical investigation of a multi-step one-shot method for frequency domain acoustic full waveform inversion

A. Sior, B. Martin and C. Geuzaine

University of Liège, Belgium

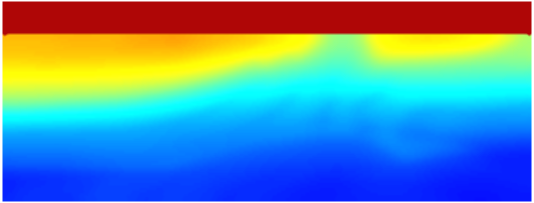
June 25, 2025

LIÈGE université

Introduction

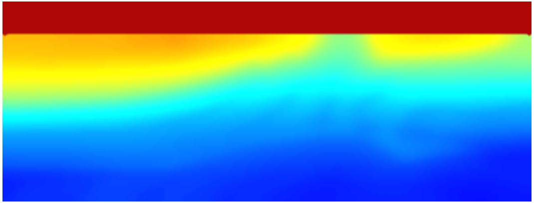
Full waveform inversion (FWI) (I)

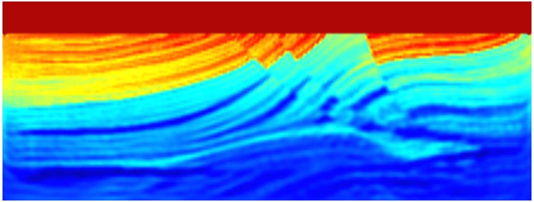
FWI is an **imaging method** that reconstructs physical properties of a sample by **minimizing** the mismatch between measured wave scattering data on the boundary of the sample and data obtained by **full-wave simulations**



Full waveform inversion (FWI) (I)

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) (II)

In the **frequency domain**, FWI of the **squared wave slowness** field m amounts to the inversion of the forward Helmholtz problem

$$\begin{cases} \Delta_x u + \omega^2 m u & = b \\ \partial_n u + \imath\omega\sqrt{m} u & = 0 \end{cases} \implies A(m)u = b$$

by minimizing an error functional J between observed data at receivers d and the waveform u induced by the squared wave slowness m

$$\arg\min_m J(u(m)) \text{ with } u(m) = u \text{ such that } A(m)u = b$$

Full waveform inversion (FWI) (II)

In the **frequency domain**, FWI of the **squared wave slowness** field m amounts to the inversion of the forward Helmholtz problem

$$\begin{cases} \Delta_x u + \omega^2 m u & = b \\ \partial_n u + \imath\omega\sqrt{m} u & = 0 \end{cases} \implies A(m)u = b$$

by minimizing an error functional J between observed data at receivers d and the waveform u induced by the squared wave slowness m

$$\arg\min_m J(u(m)) \text{ with } u(m) = u \text{ such that } A(m)u = b$$

Minimization with first- and second-order methods requires the gradient of $J(u(m))$, obtained by solving the **forward** and **adjoint** problems.

We focus on the one-shot paradigm applied to solve of the **linearized inverse problem**. Linearizing around the background slowness $\tilde{m}$ and waveform field $\tilde{u}$ such that

$$m = \tilde{m} + \delta m \text{ and } u = \tilde{u} + \delta u$$

Solving the linearized inverse problem therefore becomes finding δm such that

$$\arg\min_{\delta m} \tilde{J}(\delta u(\delta m)) \text{ with } \delta u(\delta m) = \delta u \text{ with } A(\tilde{m})\delta u = -\omega^2 \tilde{U} \delta m$$

Linearized inverse problem

We focus on the one-shot paradigm applied to solve of the **linearized inverse problem**. Linearizing around the background slowness $\tilde{m}$ and waveform field $\tilde{u}$ such that

$$m = \tilde{m} + \delta m \text{ and } u = \tilde{u} + \delta u$$

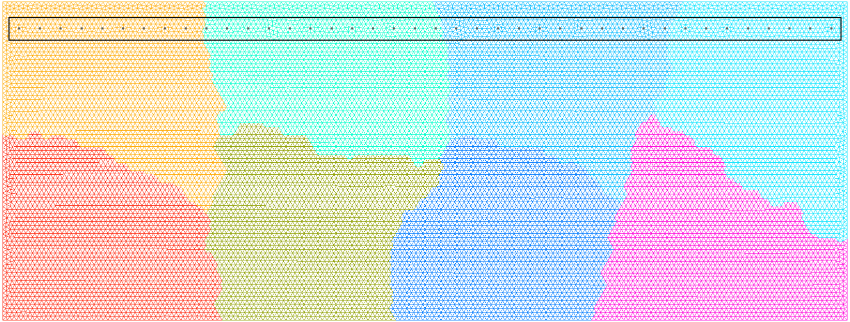
Solving the linearized inverse problem therefore becomes finding δm such that

$$\arg\min_{\delta m} \tilde{J}(\delta u(\delta m)) \text{ with } \delta u(\delta m) = \delta u \text{ with } A(\tilde{m})\delta u = -\omega^2 \tilde{U} \delta m$$

The linearized problem has a **constant system matrix** and its misfit functional is **convex**

Domain decomposition method

Computing the gradient requires solving the **forward** and **adjoint** problems. We focus on iteratively solving these linear systems using an **ORAS** preconditioner for the partitioned domain



Reference iterative FWI

When the forward and adjoint problems are solved iteratively, the inversion algorithm has the form

Reference iterative FWI

$\delta u^0 = 0; \lambda^0 = 0$
for $n = 1, 2, \dots$ **do**
  $\delta u_0^n = \delta u^{n-1}$; $\lambda_0^n = \lambda^{n-1}$
  for $\ell = 1, 2, \dots, a$ **do**
    $\delta u_\ell^n = \langle \ell\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n \rangle$
  end for
  for $\ell = 1, 2, \dots, b$ **do**
    $\lambda_\ell^n = \langle \ell\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u_a^n - \delta d) \rangle$
  end for
  $\delta u^n = \delta u_a^n$; $\lambda^n = \lambda_b^n$
  $g^n = \texttt{grad}(\delta u^n, \lambda^n)$
  $\delta m^{n+1} = \langle n\text{-th iterate of } \texttt{optimize} \rangle$
end for

Reference iterative FWI

When the forward and adjoint problems are solved iteratively, the inversion algorithm has the form

Reference iterative FWI

$\delta u^0 = 0; \lambda^0 = 0$
for $n = 1, 2, \ldots$ **do**
    $\delta u_0^n = \delta u^{n-1}; \; \lambda_0^n = \lambda^{n-1}$
    for $\ell = 1, 2, \ldots, a$ **do**
        $\delta u_\ell^n = \langle \ell$-th iterate of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n \rangle$
    end for
    for $\ell = 1, 2, \ldots, b$ **do**
        $\lambda_\ell^n = \langle \ell$-th iterate of `solve` for $\tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u_a^n - \delta d) \rangle$
    end for
    $\delta u^n = \delta u_a^n; \; \lambda^n = \lambda_b^n$
    $g^n = \mathtt{grad}(\delta u^n, \lambda^n)$
    $\delta m^{n+1} = \langle n$-th iterate of `optimize` $\rangle$
end for

Reference iterative FWI

When the forward and adjoint problems are solved iteratively, the inversion algorithm has the form

Reference iterative FWI

```
δu^0 = 0; λ^0 = 0
for n = 1, 2, ... do
    δu^n_0 = δu^{n-1}; λ^n_0 = λ^{n-1}
    for ℓ = 1, 2, ..., a do
        δu^n_ℓ = ⟨ℓ-th iterate of solve for Ãδu = −ω²Ũδm^n⟩
    end for
    for ℓ = 1, 2, ..., b do
        λ^n_ℓ = ⟨ℓ-th iterate of solve for Ã*λ = P*_Γ(P_Γ δu^n_a − δd)⟩
    end for
    δu^n = δu^n_a; λ^n = λ^n_b
    g^n = grad(δu^n, λ^n)
    δm^{n+1} = ⟨n-th iterate of optimize⟩
end for
```

Reference iterative FWI

When the forward and adjoint problems are solved iteratively, the inversion algorithm has the form

Reference iterative FWI

$\delta u^0 = 0; \lambda^0 = 0$
for $n = 1, 2, \ldots$ **do**
    $\delta u_0^n = \delta u^{n-1}; \lambda_0^n = \lambda^{n-1}$
    for $\ell = 1, 2, \ldots, a$ **do**
        $\delta u_\ell^n = \langle \ell$-th iterate of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n \rangle$
    end for
    for $\ell = 1, 2, \ldots, b$ **do**
        $\lambda_\ell^n = \langle \ell$-th iterate of `solve` for $\tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u_a^n - \delta d) \rangle$
    end for
    $\delta u^n = \delta u_a^n; \lambda^n = \lambda_b^n$
    $g^n = \texttt{grad}(\delta u^n, \lambda^n)$
    $\delta m^{n+1} = \langle n$-th iterate of `optimize` $\rangle$
end for

One-step one-shot

The one-shot paradigm consists in iterating on the forward & adjoint solvers concurrently to the minimization

<u>Reference iterative FWI</u>

$\delta u^0 = 0; \lambda^0 = 0$
for $n = 1, 2, \ldots$ **do**
  $\delta u_0^n = \delta u^{n-1}$; $\lambda_0^n = \lambda^{n-1}$
  for $\ell = 1, 2, \ldots, a$ **do**
    $\delta u_\ell^n = \langle \ell$-th iterate of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n \rangle$
  end for
  for $\ell = 1, 2, \ldots, b$ **do**
    $\lambda_\ell^n = \langle \ell$-th iterate of `solve` for $\tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u_a^n - \delta d) \rangle$
  end for
  $\delta u^n = \delta u_a^n$; $\lambda^n = \lambda_b^n$
  $g^n = \mathtt{grad}(\delta u^n, \lambda^n)$
  $\delta m^{n+1} = \langle n$-th iterate of `optimize`$\rangle$
end for

One-step one-shot

The one-shot paradigm consists in iterating on the forward & adjoint solvers concurrently to the minimization

One-step one-shot FWI

$\delta u^0 = 0; \lambda^0 = 0$

for $n = 1, 2, \ldots$ **do**

$\quad\quad\delta u^n = \langle$Next iterate of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n\rangle$

$\quad\quad\lambda^n = \langle$Next iterate of `solve` for $\tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u^{n-1} - \delta d)\rangle$

$\quad g^n = \mathtt{grad}(\delta u^n, \lambda^n)$

$\quad \delta m^{n+1} = \langle n$-th iterate of `optimize`$\rangle$

end for

The one-shot paradigm consists in iterating on the forward & adjoint solvers concurrently to the minimization

One-step one-shot FWI

1: $\delta u^0 = 0; \lambda^0 = 0$
2: **for** $n = 1, 2, \ldots$ **do**
3: $\quad \delta u^n = \langle$Next iterate of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n\rangle$
4: $\quad \lambda^n = \langle$Next iterate of `solve` for $\tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u^{n-1} - \delta d)\rangle$
5: $\quad g^n =$ `grad`$(\delta u^n, \lambda^n)$
6: $\quad \delta m^{n+1} = \langle n$-th iterate of `optimize`$\rangle$
7: **end for**

When the system solver is the **stationary iteration** and the minimization scheme is **fixed-step gradient descent**, this converges under some conditions (Bonazzoli, Haddar and Vu, 2022)

Multi-step one-shot

Multi-step one-shot increases the amount of forward & adjoint solver iterations done per minimization step and limits them to k

Multi-step one-shot FWI

$\delta u^0 = 0; \lambda^0 = 0$
for $n = 1, 2, \ldots$ **do**
  $\delta u_0^n = \delta u^{n-1};\ \lambda_0^n = \lambda^{n-1}$
  for $\ell = 1, 2, \ldots, \boxed{k}$ **do**
    $\delta u_\ell^n = \langle \ell\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}\delta u = -\omega^2 \tilde{U} \delta m^n \rangle$
  end for
  for $\ell = 1, 2, \ldots, \boxed{k}$ **do**
    $\lambda_\ell^n = \langle \ell\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u_k^n - \delta d) \rangle$
  end for
  $\delta u^n = \delta u_k^n;\ \lambda^n = \lambda_k^n$
  $g^n = \texttt{grad}(\delta u^n, \lambda^n)$
  $\delta m^{n+1} = \langle n\text{-th iterate of } \texttt{optimize} \rangle$
end for

Fixed-step gradient descent

Stationary iteration: convergence map (I)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,\mathrm{Hz}$

- 100 multi-step one-shot **gradient descent iterations** with a **stationary forward & adjoint solver** are run with a given k and τ

Stationary iteration: convergence map (I)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,\mathrm{Hz}$

- 100 multi-step one-shot **gradient descent iterations** with a **stationary forward & adjoint solver** are run with a given k and τ
- Convergence is deemed to be reached at iteration i such that

$$\frac{|g^n|}{\max_N |g^N|} < 0.1 \text{ for all } n \geq i,$$

  where g is the gradient

Stationary iteration: convergence map (I)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,\mathrm{Hz}$

- 100 multi-step one-shot **gradient descent iterations** with a **stationary forward & adjoint solver** are run with a given k and τ
- Convergence is deemed to be reached at iteration i such that

$$\frac{|g^n|}{\max_N |g^N|} < 0.1 \text{ for all } n \geq i,$$

  where g is the gradient
- The cost is the number of **ORAS preconditioner applications** for 8 subdomains

Stationary iteration: convergence map (I)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,$Hz

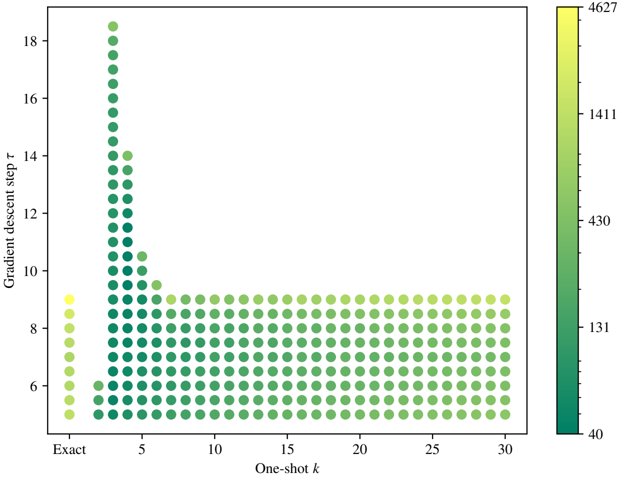
- 100 multi-step one-shot **gradient descent iterations** with a **stationary forward & adjoint solver** are run with a given k and τ
- Convergence is deemed to be reached at iteration i such that

$$\frac{|g^n|}{\max_N |g^N|} < 0.1 \text{ for all } n \geq i,$$

  where g is the gradient
- The cost is the number of **ORAS preconditioner applications** for 8 subdomains
- This cost is compared with the **reference iterative FWI** with a relative residual criterion of 10^{-6} for solves

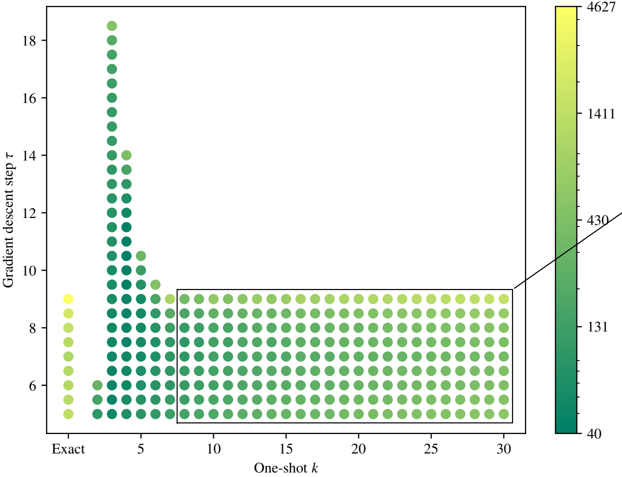
Stationary iteration: convergence map (II)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,\text{Hz}$



Stationary iteration: convergence map (II)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,\text{Hz}$

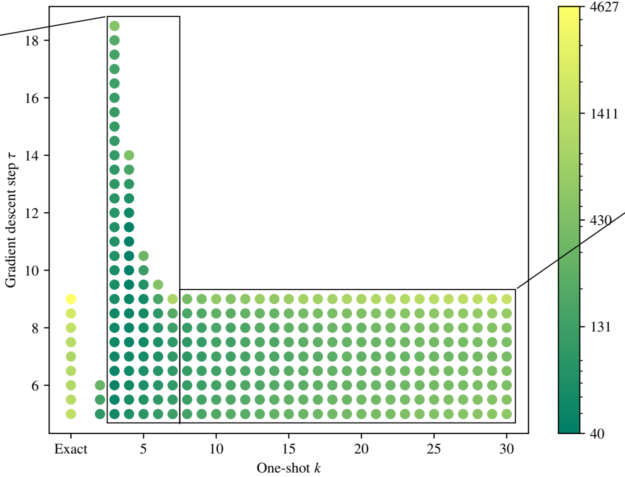


Minimization behaves like the reference

Stationary iteration: convergence map (II)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,$Hz

Maximal allowable step size $\tau_{\max}$ has a significant peak above the reference gradient regime

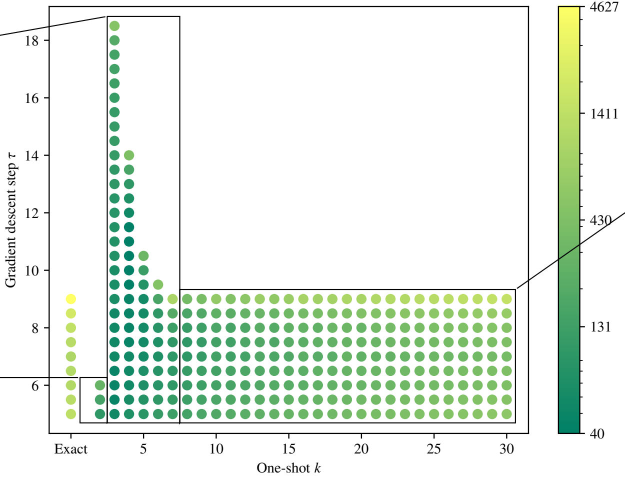


Minimization behaves like the reference

Stationary iteration: convergence map (II)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at 1 Hz

Maximal allowable step size $\tau_{\max}$ has a significant peak above the reference gradient regime



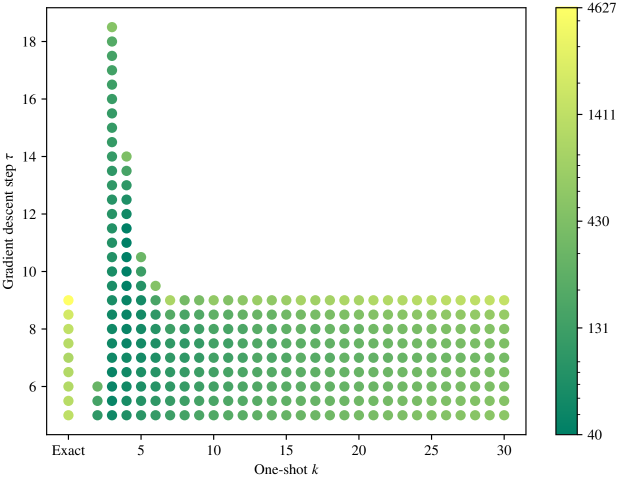
Maximal allowable step size $\tau_{\max}$ drops

Minimization behaves like the reference

Stationary iteration: convergence map (II)

The convergence behavior of the multi-step one-shot algorithm is mapped as a function of k and the step size τ for the Marmousi reference model at $1\,$Hz

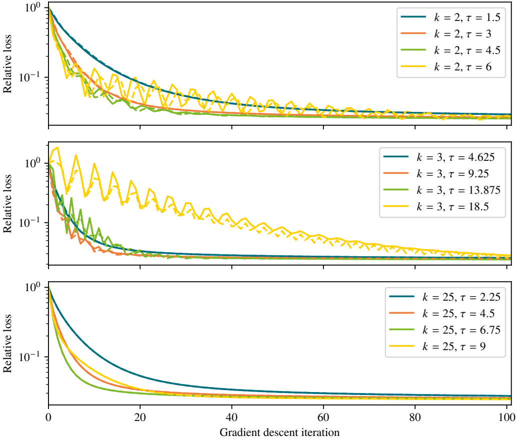
For a given step size τ, choosing an adequate value of k is **effective** at reducing the cost to reach convergence, even if this value is within the intermediate range



Stationary iteration: relative loss

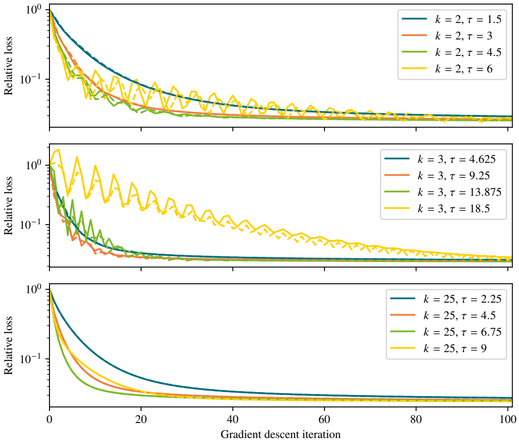
The behavior relative loss $J(\delta u(\delta m))$ is monitored as τ approaches its maximal allowable value for a given k

- For $k = 25$, the convergence remains monotonous as τ approaches $\tau_{\max}$



The behavior relative loss $J(\delta u(\delta m))$ is monitored as τ approaches its maximal allowable value for a given k

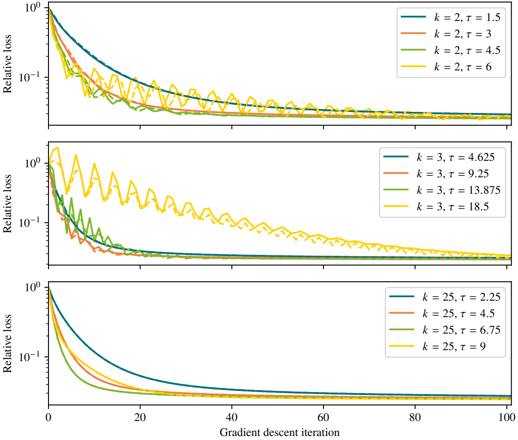
- For $k = 25$, the convergence remains monotonous as τ approaches $\tau_{\max}$
- For $k = 2$ and 3, non-monotonous yet convergent behavior appears well below $\tau_{\max}$



Stationary iteration: relative loss

The behavior relative loss $J(\delta u(\delta m))$ is monitored as τ approaches its maximal allowable value for a given k

- For $k = 25$, the convergence remains monotonous as τ approaches $\tau_{\max}$
- For $k = 2$ and 3, non-monotonous yet convergent behavior appears well below $\tau_{\max}$
- Although one-shot grants higher τ-robustness for $k = 3$ here, this does not necessarily translate to faster convergence

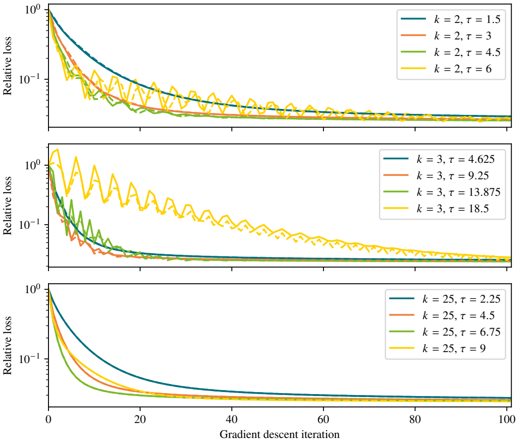


Stationary iteration: relative loss

The behavior relative loss $J(\delta u(\delta m))$ is monitored as τ approaches its maximal allowable value for a given k

- For $k = 25$, the convergence remains monotonous as τ approaches $\tau_{\max}$
- For $k = 2$ and 3, non-monotonous yet convergent behavior appears well below $\tau_{\max}$
- Although one-shot grants higher τ-robustness for $k = 3$ here, this does not necessarily translate to faster convergence

Overall, one-shot "loosens" the limit between monotonous convergence and diverging oscillations by facilitating **oscillating convergent** behaviors



Stationary iteration: gradient evolution with k (I)

To obtain an intuitive understanding, we compare the k-step one-shot estimated gradient g is compared to the LU-computed reference gradient g_{ref}

- 100 k-step one-shot **gradient descent iterations** with a **direct forward & adjoint solver** for a given τ

Stationary iteration: gradient evolution with k (I)

To obtain an intuitive understanding, we compare the k-step one-shot estimated gradient g is compared to the LU-computed reference gradient g_{ref}

- 100 k-step one-shot **gradient descent iterations** with a **direct forward & adjoint solver** for a given τ
- At each step, a **one-shot estimation** of the gradient is computed for a given k, using a **stationary forward & adjoint solver**

Stationary iteration: gradient evolution with k (I)

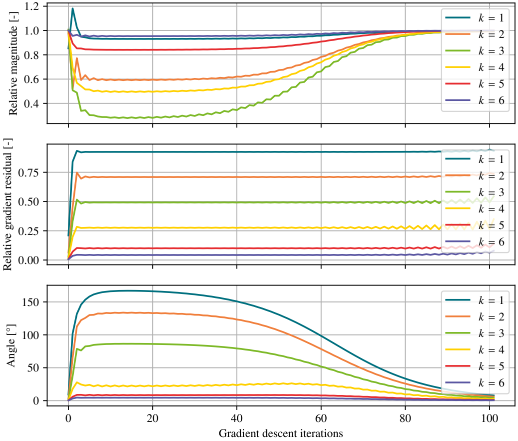
To obtain an intuitive understanding, we compare the k-step one-shot estimated gradient g is compared to the LU-computed reference gradient g_{ref}

- 100 k-step one-shot **gradient descent iterations** with a **direct forward & adjoint solver** for a given τ
- At each step, a **one-shot estimation** of the gradient is computed for a given k, using a **stationary forward & adjoint solver**
- This one-shot gradient is compared in **relative magnitude**, **angle**, and **relative difference** with the reference LU gradient

Stationary iteration: gradient evolution with k (II)

This experiment is performed on the linearized inverse problem at $1\,\text{Hz}$ using the Marmousi case (8 subdomains, METIS) for $\tau = 9$

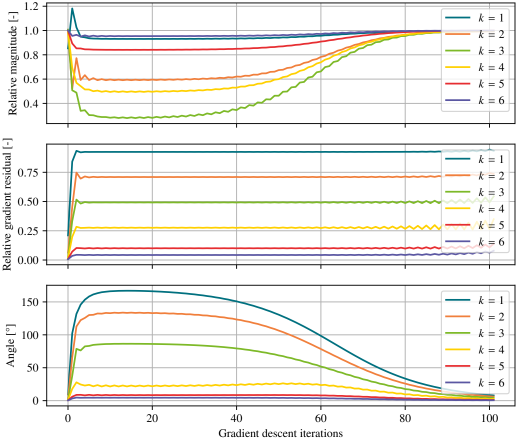
- As k increases, the **angle** between the estimated and reference gradients **monotonously decreases**



Stationary iteration: gradient evolution with k (II)

This experiment is performed on the linearized inverse problem at $1\,\text{Hz}$ using the Marmousi case (8 subdomains, METIS) for $\tau = 9$

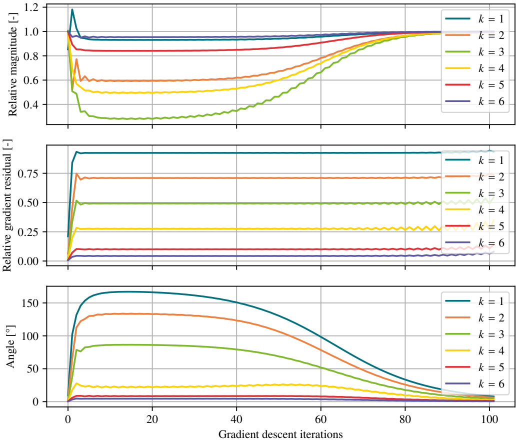
- As k increases, the **angle** between the estimated and reference gradients **monotonously decreases**
- The relative magnitude first **decreases** then **increases** as it approaches the reference gradient



Stationary iteration: gradient evolution with k (II)

This experiment is performed on the linearized inverse problem at $1\,\mathrm{Hz}$ using the Marmousi case (8 subdomains, METIS) for $\tau = 9$

- As k increases, the **angle** between the estimated and reference gradients **monotonously decreases**
- The relative magnitude first **decreases** then **increases** as it approaches the reference gradient
- The relative gradient difference is **roughly constant** for a given k



Adaptive step one-shot

Barzilai-Borwein method

Step size adaptivity for the descent can be implemented with the **Barzilai-Borwein** method

- The method introduces a **short** and a **long** step size

$$p^n_{\text{short}} = -\frac{\langle \Delta\delta m, \Delta g \rangle}{\langle \Delta g, \Delta g \rangle} g^n \quad \text{and} \quad p^n_{\text{long}} = -\frac{\langle \Delta\delta m, \Delta\delta m \rangle}{\langle \Delta\delta m, \Delta g \rangle} g^n$$

Barzilai-Borwein method

Step size adaptivity for the descent can be implemented with the **Barzilai-Borwein** method

- The method introduces a **short** and a **long** step size

$$p^n_{\text{short}} = -\frac{\langle \Delta\delta m, \Delta g\rangle}{\langle \Delta g, \Delta g\rangle} g^n \quad \text{and} \quad p^n_{\text{long}} = -\frac{\langle \Delta\delta m, \Delta\delta m\rangle}{\langle \Delta\delta m, \Delta g\rangle} g^n$$

- The geometric **mean** of the long and short step size is also useful in non-convex scenarios

$$p^n_{\text{mean}} = -\frac{|\Delta\delta m|}{|\Delta g|} g^n$$

Barzilai-Borwein method

Step size adaptivity for the descent can be implemented with the **Barzilai-Borwein** method

- The method introduces a **short** and a **long** step size

$$p^n_{\text{short}} = -\frac{\langle \Delta\delta m, \Delta g\rangle}{\langle \Delta g, \Delta g\rangle} g^n \quad \text{and} \quad p^n_{\text{long}} = -\frac{\langle \Delta\delta m, \Delta\delta m\rangle}{\langle \Delta\delta m, \Delta g\rangle} g^n$$

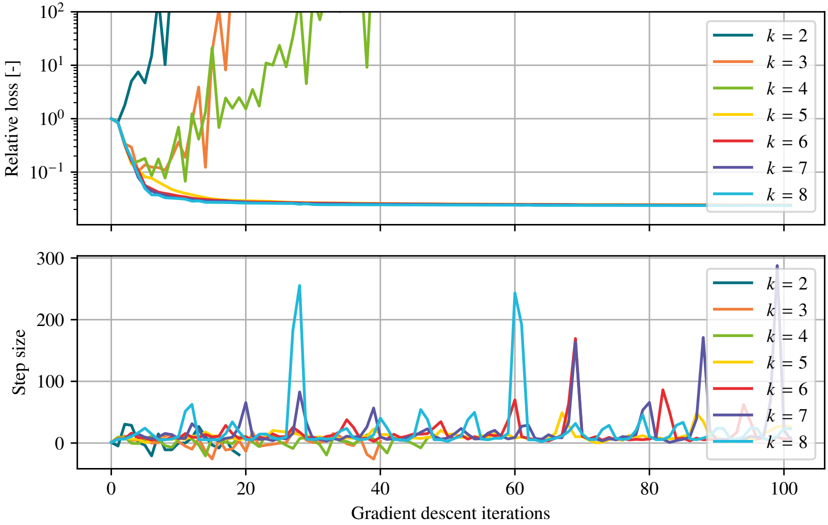
- The geometric **mean** of the long and short step size is also useful in non-convex scenarios

$$p^n_{\text{mean}} = -\frac{|\Delta\delta m|}{|\Delta g|} g^n$$

- For minimization of convex functionals, the short and long step sizes converge, albeit non-monotonously (Raydan, 1991)

Barzilai-Borwein: short step relative losses

The short and mean step sizes were found to be robust to the one-shot estimated gradients



Adaptive k one-shot

Negative step ramp-up

Requirement that the Barzilai-Borwein step size be positive. A strategy to upgrade k is therefore to

- Set k to an initial value of k_0

Negative step ramp-up

Requirement that the Barzilai-Borwein step size be positive. A strategy to upgrade k is therefore to

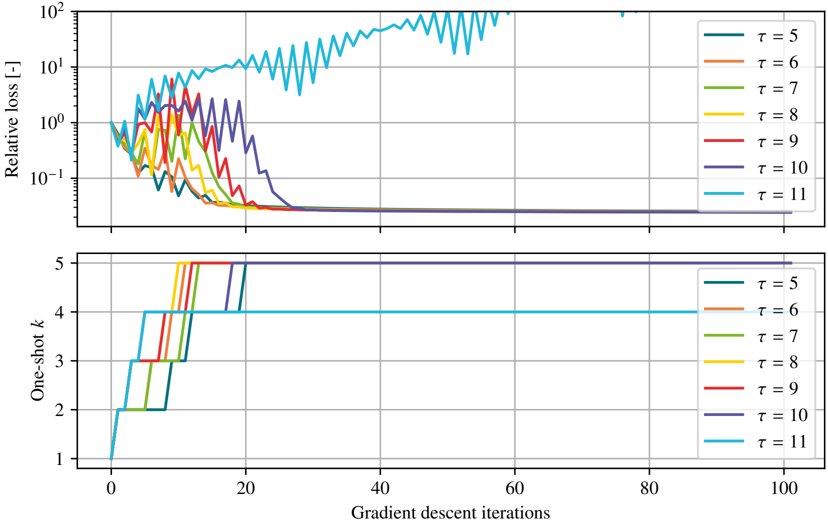
- Set k to an initial value of k_0
- Increase it every time the value of

$$\langle \Delta\delta m, \Delta g \rangle$$

is **negative**, as sampled by the one-shot estimated gradients

Negative step ramp-up: relative losses and k

This strategy was tested for the inversion of the linearized inverse problem at $1\,$Hz on the Marmousi model (8 subdomains, METIS) for various step sizes τ, with $k_0 = 1$



Non-linearized inverse problem

Gauss-Newton one-shot

The non-linearized inverse problem can be solved as a succession of linearized inverse problems using the Gauss-Newton method. A globalized scheme for the complete problem optimization becomes

1. Find the step direction p, the solution of the locally linearized problem

$$p = \arg\min_{\delta m} \tilde{J}(u(\delta m))$$

2. Find a scaling factor α that enforces the Armijo-Goldstein condition

$$J(u(m_k + \alpha p)) \leq J(u(m_k)) + c_1 D_m J(u(m_k))(\alpha p)$$

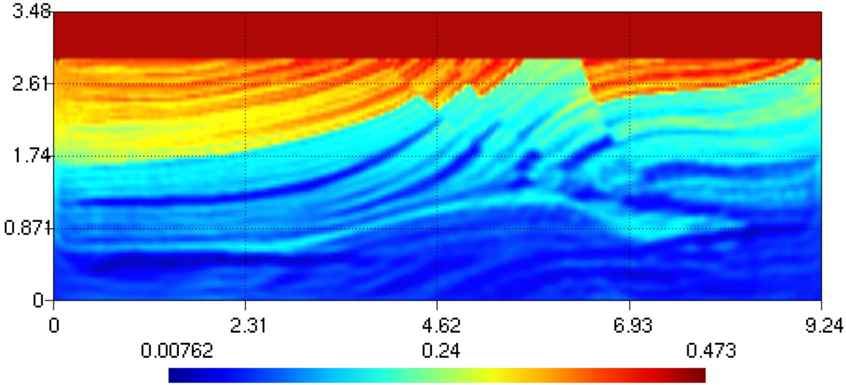
3. Update m such that

$$m_{k+1} = m_k + \alpha p$$

Step 1 may be done using one-shot gradient descent with adaptive step sizes and k schemes

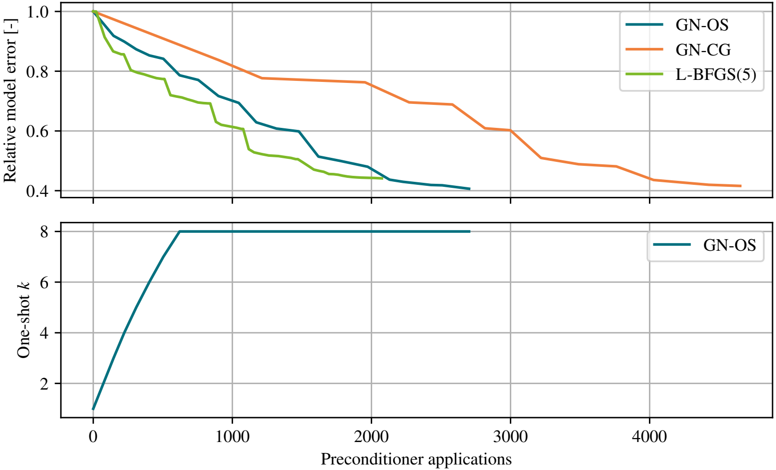
Test case 1: Marmousi (I)

This scheme was tested for the sequential inversion of the Marmousi case (Versteeg, 1994) at frequencies $\{1, 1.6, 2.5, 4, 6.5, 10.4\}$ Hz



Test case 1: Marmousi (II)

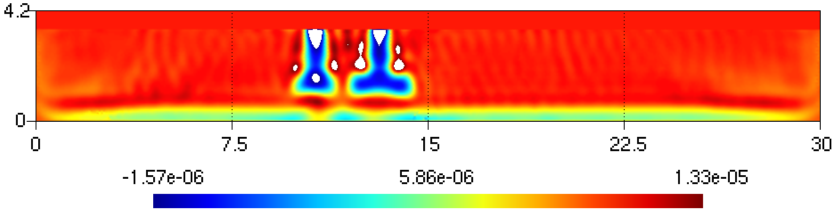
The inversion is compared with L-BFGS(5) and the Gauss-Newton-CG



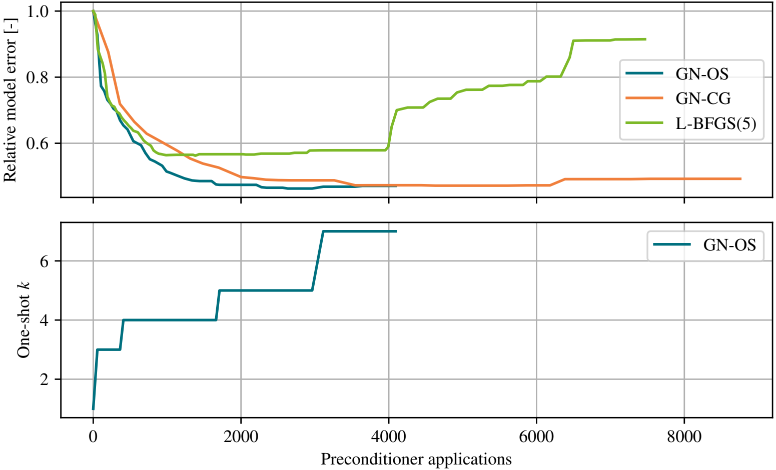
Using one-shot optimization to solve the Gauss-Newton equation leads to performances comparable to that of state-of-the-art algorithms

Test case 2: T-shaped reflectors (I)

This scheme was also tested for the inversion of near-surface imaging application consisting of T-shaped reflectors in a soil (Métivier et al., 2017) (with frequencies $\{100, 125, 150, 175, 200, 225, 250, 275, 300\}$ Hz)



Test case 2: T-shaped reflectors (II)



Conclusions & perspectives

Conclusions & perspectives

Coupling forward & adjoint iterations with the optimization is an interesting avenue for potential performance gains when solving inverse problems:

- Gauss-Newton one-shot seems competitive with state-of-the-art methods!

Conclusions & perspectives

Coupling forward & adjoint iterations with the optimization is an interesting avenue for potential performance gains when solving inverse problems:

- Gauss-Newton one-shot seems competitive with state-of-the-art methods!
- There is a **trade-off** between the **estimated gradient accuracy** and the **complexity** of descent methods that can be employed

Conclusions & perspectives

Coupling forward & adjoint iterations with the optimization is an interesting avenue for potential performance gains when solving inverse problems:

- Gauss-Newton one-shot seems competitive with state-of-the-art methods!
- There is a **trade-off** between the **estimated gradient accuracy** and the **complexity** of descent methods that can be employed
- Using a succession of linearized inverse problems to solve the complete inverse problem is a bottleneck due to the need to compute $\tilde{u}$.

Conclusions & perspectives

Coupling forward & adjoint iterations with the optimization is an interesting avenue for potential performance gains when solving inverse problems:

- Gauss-Newton one-shot seems competitive with state-of-the-art methods!
- There is a **trade-off** between the **estimated gradient accuracy** and the **complexity** of descent methods that can be employed
- Using a succession of linearized inverse problems to solve the complete inverse problem is a bottleneck due to the need to compute $\tilde{u}$.
- Applying one-shot on the non-linearized problem remains to be explored
  - $\rightarrow$ No convexity guarantee
  - $\rightarrow$ Need a one-shot compatible globalization strategy

Conclusions & perspectives

Coupling forward & adjoint iterations with the optimization is an interesting avenue for potential performance gains when solving inverse problems:

- Gauss-Newton one-shot seems competitive with state-of-the-art methods!
- There is a **trade-off** between the **estimated gradient accuracy** and the **complexity** of descent methods that can be employed
- Using a succession of linearized inverse problems to solve the complete inverse problem is a bottleneck due to the need to compute $\tilde{u}$.
- Applying one-shot on the non-linearized problem remains to be explored
  - $\rightarrow$ No convexity guarantee
  - $\rightarrow$ Need a one-shot compatible globalization strategy
- The choice of an adequate k remains elusive

Contact

✉ alejandro.sior@student.uliege.be

Access my masters' thesis: