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

Author
Alejandro W. Sior

Supervised by
Prof. Christophe Geuzaine

Master's thesis completed in order to obtain the degree of
Master of Science in Aerospace Engineering

University of Liège
School of Engineering and Computer Science

Academic year 2024-2025

ABSTRACT

Full waveform inversion is an imaging method consisting of the spatial reconstruction of the parameter field of a wave propagation problem through the optimization of a misfit functional between the synthetic solution of the problem and the observed solution. Full waveform inversion is typically articulated in two distinct layers: the inner layer involves solving the forward and adjoint problems for the calculation of the misfit gradient, and the outer layer performing the optimization. For large scale problems, the forward and adjoint problems need to be solved using an iterative domain decomposition method.

The one-shot paradigm couples these iterative solutions with the optimization steps by limiting the number of forward and adjoint solver iterations to a small number, thus producing an inexact gradient.

In this master's thesis, the practical feasibility and performance usefulness of the one-shot paradigm are investigated in the context of full waveform inversion in the frequency domain. Inversion with a variant of the one-shot algorithm proposed by Bonazzoli, Haddar, and Vu (2022) with gradient descent for minimization and an Optimized Restricted Additive Schwarz-preconditioned stationary iterative solver for the forward and adjoint problems is tested, and its cost is measured in terms of the number of forward and adjoint iterations. Then, a Gauss-Newton method based on the one-shot paradigm and an adaptive gradient descent is introduced and compared with other algorithms.

For the inversion of the linearized inverse problem, the results indicate that the multi-step one-shot paradigm may allow converging with a reduced number of forward and adjoint solver iterations for a given step size, while increasing step size robustness. The Barzilai-Borwein method is introduced for step size adaptiveness and is shown to be robust to one-shot estimated gradients, provided that sufficient forward and adjoint iterations are performed. The resulting Gauss-Newton algorithm for the inversion of the full inverse Helmholtz problem shows comparable performances to state-of-the-art methods.

EXAMINATION COMMITTEE

Supervisor	Prof. Christophe Geuzaine
Examiners	Prof. Maarten Arnst
	Boris Martin
	Prof. Frédéric Nguyen

ACKNOWLEDGEMENTS

This work would not have been possible without the support of Prof. Christophe Geuzaine and Boris. Thank you both for always being there to address my questions and doubts, for your valuable advice, and for guiding me throughout this journey.

I would like to thank all the members of the Applied and Computational Electromagnetics group for their warm welcome and encouragement during the completion of this work. Thanks to Boris, Florent, and Quentin for accommodating me in their office space.

Many thanks to the examiners of this master's thesis: Boris, Prof. Maarten Arnst and Prof. Frédéric Nguyen, for agreeing to evaluate my work. Once again, I thank Prof. Christophe Geuzaine and Boris for their valuable review and remarks.

Louise, thank you for your constant support and for all the love and moments we shared during this semester.

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Contents

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

Alejandro W. Sior

Introduction

Inverse problems are a class of problems in which a parameter m involved in the problem is reconstructed using the complete or partial knowledge of the observed solution d [36]. Various types of models may be inverted to obtain a qualitative and quantitative understanding of the unobservable variables that govern the behavior of the model.

Specifically, in the context of models governed by a partial differential equation, consider the general problem defined on the domain Ω and its boundary $\partial\Omega$, which consists of finding u defined on Ω such that

$$\begin{cases} A(m,u) = 0 & \text{in } \Omega, \\ g(m,u) = 0 & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where A and g are the differential equation and boundary condition operators, and m is the parameter of the problem. Problem (1) constitutes the *direct* problem, and its solution u is the synthetic solution of the problem. In the frame of problem (1), consider now that an observed solution d is known on a subdomain Γ of Ω; a subsequent problem would be to let m be a degree of freedom and find m such that

$$\arg\min_m J(u) \text{ with } u \text{ such that } \begin{cases} A(m,u) = 0 & \text{in } \Omega, \\ g(m,u) = 0 & \text{on } \partial\Omega, \end{cases} \tag{2}$$

where J is a misfit functional measuring the difference between u and d on Γ. Problem (2) constitutes an *inverse* problem, and its solution is m, the parameter of the direct problem such that the synthetic solution approximates the observed solution on Γ. A typical choice of misfit function is often related to the squared norm of the difference between u and d on Γ.

The classical methods employed to solve inverse problems such as (2) are based around numerical optimization. The workflow can be articulated in two layers. The outer layer of inversion consists in the use of an optimization algorithm [36], such as gradient descent, quasi-Newton methods, or inexact Newton to minimize the functional [2]. As these methods require knowing the gradient of J as a function of m, the inner layer consists in calculating the gradient, which can be done using finite-differences on the direct problem or using the adjoint-state method. These inner layer gradient calculation methods involve solving the direct problem (or the related adjoint problem), which is done using direct or iterative methods.

An important subclass of inverse problems of type (2) in geophysics and biomedical imaging is one of imaging the physical properties of a medium based on the wave propagation patterns

measured at a set of locations. Specifically, a boundary setup in which a set of emitters successively emit waves, which are then received by a set of receivers, can be employed to acquire the data necessary to perform the imaging of the density, the wave speed, or the Lamé parameters of the medium. This process is called Full Waveform Inversion (FWI) and it has seen successful applications in acoustic [19], [25] and electromagnetic [17], [20] imaging. The core problem governing the wave propagation in the medium depends on the underlying physics of the problem as well as modelling assumptions. In the case of acoustic problems or elastic problems for which the assumption of single wave propagation speed is done, the governing partial differential equation, posed in the Fourier domain, is the Helmholtz equation [36].

When solving large inverse Helmholtz problems, one scalable way to efficiently organize the inner layer is by decomposing the domain Ω into several subdomains and to parallelize the solutions of the forward and adjoint problems on several processes using an (Optimized) Restricted Additive Schwarz ((O)RAS) preconditioner [10], [34] for Krylov subspace iterative methods. Methods built upon this paradigm fall into the realm of domain decomposition methods (DDMs).

Work carried out by Bonazzoli et al. prompted interest into a new organization paradigm of the outer and inner layers [7]. Instead of solving the inner layer direct (or adjoint) problems down to the stopping criterion of the iterative methods, only approximate solutions are sought by performing a limited amount of iterations of the chosen iterative method. This results in the outer layer optimization and the inner layer direct problem solutions being calculated quasi-concurrently instead of sequentially. These approximations of the direct (or adjoint) problem solutions make the resulting gradient evaluation approximate; the key idea is that when the optimization algorithm is gradient descent, the forward and adjoint solver is the stationary iteration, and granted an adequate choice of the inner iterations count, the method has been shown to converge when the descent step size is sufficiently small. Also, in simple cases, the method has been empirically shown to converge in *fewer* iterations than solving the inner layer problems fully, hinting at potential performance gains [37]. This paradigm is called the *one-shot* method.

The purpose of this master's thesis is to investigate the addition of one-shot techniques into an existing parallelized iterative inverse Helmholtz solver, developed in the Applied and Computational Electromagnetics group at the University of Liège [22]. The core of this work can be articulated along two axes. The first axis consists of the inclusion of one-shot techniques into the solver, and the assessment of the viability of those techniques for a reference geophysical benchmark in the case of a linearized inverse Helmholtz problem. The second axis is to investigate how these methods can be extended into a practical algorithm for solving a non-linearized inverse Helmholtz problem with multiple frequencies.

Chapter 1

General background

The inverse Helmholtz solver used as a basis for the inclusion of one-shot methods was developed by Boris Martin as part of his PhD thesis. This solver is a two-dimensional FWI solver that optimizes the complete inverse problem or the locally linearized inverse problem and uses either direct or iterative methods to solve the direct and adjoint problems. When iterative methods are used to solve the inner layer problems, the solver makes use of ORAS preconditioners to parallelize the solution across subdomains. The optimization of the complete problem is performed using gradient descent, L-BFGS or Gauss-Newton. When using Gauss-Newton optimization, the locally linearized inverse problem is solved using gradient descent or conjugate gradients.

The purpose of this chapter is to present the mathematical background behind the base solver and state-of-the-art FWI techniques.

1.1 The Helmholtz equation

The Helmholtz equation is the equation that describes the steady state complex waveform of angular frequency ω induced by source terms h in an isotropic medium of wave squared slowness m. The equation is written as

$$\Delta_x u + \omega^2 m u = h, \tag{1.1}$$

where u is the waveform field.

Intuitive understanding of (1.1) can be obtained by its derivation from the wave equation. Consider the propagation of a wave $\hat{u}$, induced by a source field $\hat{h}$, in an isotropic medium, where the speed of propagation c is a function of the position within the medium. The dynamics of the propagation are governed by the equation

$$\frac{\partial^2 \hat{u}}{\partial^2 t} - c^2 \Delta_x \hat{u} = \hat{h}. \tag{1.2}$$

Assuming a steady harmonic oscillation, $\hat{u}$ may be written as a product of a temporal component, namely the harmonic oscillator $\exp(\imath\omega t)$ and a complex spatial component that describes the form of the oscillation in space; this separation of variables is

$$\hat{u}(t, x) = \mathrm{Re}\left\{u(x) \exp(\imath\omega t)\right\}. \tag{1.3}$$

Inserting (1.3) into (1.2), dividing both sides by $-c^2 \exp(\imath\omega t)$ and renaming the right-hand side leads to the Helmholtz equation (1.1).

The decomposition (1.3) tells us that the solution of the Helmholtz equation u corresponds to the spatial form of the wave of angular frequency ω, its *waveform*, which is to be propagated through time by multiplying it with the complex exponential $\exp(\imath\omega t)$.

1.2 The direct Helmholtz problem

The Helmholtz equation (1.1) is a linear partial differential equation which is of second order in space. The problem of finding the waveform in a domain $\Omega \subset \mathbb{R}^n$ using the Helmholtz equation can be posed by associating the equation with boundary conditions on $\partial\Omega$.

The direct Helmholtz problem considered here consists of finding u in $C^2(\Omega)$, the space of twice continuously differentiable functions on Ω, such that

$$\begin{cases} \Delta_x u + \omega^2 m u = h & \text{in } \Omega, \\ \partial_n u + \imath\omega\sqrt{m}u = 0 & \text{on } \partial\Omega. \end{cases} \tag{1.4}$$

The operator ∂_n designates the directional derivative in the direction of the unit normal of $\partial\Omega$. Proper care must be taken in defining the boundary conditions involved in the problem (1.4), as they are an important aspect of the practical frame in which the problem is set. In this example, we consider a class of imaging scenarios in which the domain Ω of interest is a subdomain of an infinite medium; the boundary condition at the interface between Ω and the surrounding medium must then model the seamless transmission of the waves in and out of Ω without partial reflection. This contrasts with other types of imaging scenarios, where some portion of $\partial\Omega$ interfaces with a medium which is assumed to not transmit the waves (*e.g.* the interface between the ground and the air, in the case of acoustic imaging of a ground).

The *absorption boundary condition* on $\partial\Omega$ interfacing with the surrounding medium is the second equation of (1.4). This condition is reminiscent of the transport equation in the direction normal to the boundary, taken in the Fourier domain. The transport equation in the direction normal to the boundary has the property of not reflecting any wave travelling in a direction normal to the boundary. Note that this low order transmission boundary condition is not sufficient to model the complete absorption of waves accurately; in practice, higher order transmission conditions are necessary. The modelization of such conditions in a finite element context is addressed in [29].

The source term is applied at a discrete point σ within Ω. The source is written as

$$h(x) = \delta_\sigma(x) \quad \text{at } \sigma, \tag{1.5}$$

where δ_σ is the Dirac delta centered at σ. Figure 1.1 summarizes the domain on which the problem is defined.

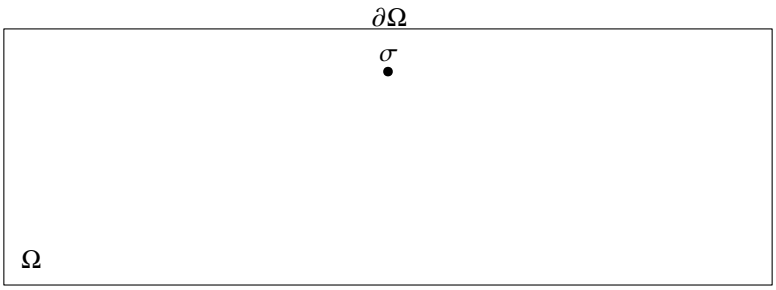


Figure 1.1: Domain and boundary on which the problem (1.4) is defined. $\partial\Omega$ is the transmission boundary and σ is the source location.

1.2.1 Weak form

The weak form of the equation can be obtained by taking the duality pairing of the equation with a function v in V which is *a priori* arbitrary, such that

$$\langle \Delta_x u + \omega^2 m u, v \rangle = \langle \delta_\sigma, v \rangle, \tag{1.6}$$

for all v in V. The left-hand side pairing is equivalent to an inner product $V \times V \to \mathbb{C}$, and the right-hand side is equal to $v(\sigma)$ by definition of Dirac delta distribution. The weak form of the equation is developed using the divergence theorem as

$$-\langle \nabla_x u, \nabla_x v \rangle + (\partial_n u, v) + \langle \omega^2 m u, v \rangle = \langle \delta_\sigma, v \rangle, \tag{1.7}$$

for all v in V, where $(\cdot, \cdot)$ is the inner product on $\partial\Omega$.

The weak form of the problem can be formulated by particularizing (1.7) to the boundary conditions and by specifying the spaces in which u and v lie. The weak form of the Helmholtz problem consists of finding u in V such that

$$-\langle \nabla_x u, \nabla_x v \rangle + (-\imath\omega\sqrt{m} u, v) + \langle \omega^2 m u, v \rangle = \langle \delta_\sigma, v \rangle, \tag{1.8}$$

for all v in V, with $V \coloneqq H^1(\Omega)$, the Sobolev space of functions on Ω with square integrable values and first weak derivatives. The requirement on m is that it is bounded and positive on Ω.

1.2.2 Discretization

The problem is discretized to obtain a system of equations which is solvable numerically. The purpose of discretization is to seek an approximation u^h in a known finite dimensional space V^h, of the exact solution u in V. Although several spatial discretization paradigms are adequate for FWI, the reference solver makes use of the finite element method (FEM). For the sake of completeness, the methodology of the FEM applied to the direct Helmholtz problem shall be presented here; foundational work and definitions of the FEM can be found in [38].

Consider the weak form (1.8) of the problem. Let V^h be a subspace of V. Since the equation holds for all v in V, it must also hold for any v^h in V^h. The approximation u^h in V^h of the solution u is obtained with a Galerkin projection by finding u^h in V^h such that

$$-\langle \nabla_x u^h, \nabla_x v^h \rangle + (-\imath\omega\sqrt{m} u^h, v^h) + \langle \omega^2 m u^h, v^h \rangle = \langle \delta_\sigma, v^h \rangle, \tag{1.9}$$

for all v^h in V^h. The left-hand side is a bilinear form $a(u^h, v^h)$, and using the Riesz representation theorem, there exists a linear application $A : V^h \to V^h$ such that $a(u^h, v^h) = \langle Au^h, v^h \rangle$ for all u^h, v^h in V^h. Likewise, the right-hand side may be written as a continuous linear functional $F(v^h)$, and there exists a unique b in V^h such that $F(v^h) = \langle b, v^h \rangle$ for all v^h in V^h. Since $Au^h - b$ is in V^h and the solution u^h satisfies $\langle Au^h - b, v^h \rangle = 0$ for all v^h in V^h, the problem (1.9) is equivalently formulated as finding u^h in V^h such that

$$Au^h = b. \tag{1.10}$$

In this work, we use first order Lagrange finite elements as a strategy to build the subspace V^h in which the approximation is sought. In the FEM, the domain is divided into elements, which are defined with a set of nodes, indexed by i, and located at positions x_i. The subspace V^h is generated by a Lagrange polynomial basis of shape functions

$$V^h = \langle \phi_1, \phi_2, ..., \phi_N \rangle, \tag{1.11}$$

where the shape functions ϕ_i are defined such that $\phi_i(x_i) = 1$ for all i and $\phi_i(x_j) = 0$ for all (i, j) with $i \neq j$, and such that they have a local support on the elements adjacent to the node. The approximate solution u^h in V^h is expressed in the basis of the shape functions with

$$u^h(x) = \sum_{j=1}^{N} \phi_j(x) u_j. \tag{1.12}$$

In what follows, the discretized solution u^h is denoted u to make the notations lighter.

1.3 Iterative methods

Given the discretized operators and fields obtained in the previous section, the discretized Helmholtz problem can be practically solved by a computer by finding u in V^h such that

$$Au = b. \tag{1.13}$$

For this, several types of methods can be used. *Direct methods* are methods based on algorithms that employ a finite number of steps to obtain a solution; these methods include substitution, Gaussian Elimination, LU decomposition, etc. The other kind of methods are called *iterative methods*, which iteratively approximate the solution.

In this work, domain decomposition methods are used to solve the equation. These methods consist of *preconditioning* the system by solving a subdomain-local problem, then by accumulating all the subdomain-local waveform fields into a domain-wide one. This preconditioner is then used with an iterative to improve its scalability. A brief overview of three iterative methods of interest are presented in this subsection.

Let us consider the usage of iterative methods to solve the following linear system

$$Ax = b; \tag{1.14}$$

iterative methods work by maintaining the k-th iterate, denoted x_k, and updating it based on some rule specific to the method and its parameters.

1.3.1 Richardson

The modified Richardson iteration, also called the stationary iteration, consists of updating the running value of the iterate using some multiple of the running value of the residual, *i.e.*

$$x_{k+1} = x_k + \alpha(b - Ax_k), \tag{1.15}$$

where α is a scalar which is a parameter of the equation. The update equation (1.15) corresponds to Algorithm 1. Algorithm 1 converges to the exact equation if and only if all the eigenvalues of $I - \alpha A$ are strictly less than 1 in absolute value, or in other words, if and only if

$$\rho(I - \alpha A) < 1. \tag{1.16}$$

Algorithm 1 Richardson method to solve the linear system $Ax = b$.

Require: x_0
  $r_0 = b - Ax_0$
  for $k = 0, 1, 2, \ldots$ **do**
    $x_{k+1} = x_k + \alpha r_k$
    $r_{k+1} = r_k - \alpha A r_k$
  end for

1.3.2 Minimized Residual

The Minimized Residual (MR) method [31], sometimes referred to as self-scaled Richardson method [3], is an iterative method that, like the Richardson iteration, steps in the direction of the running residual, *i.e.*

$$x_{k+1} = x_k + \alpha_k(b - Ax_k), \tag{1.17}$$

but it differs in that the scaling factor α_k is updated at each iteration. The value of α_k is calculated such that the next value of the residual r_{k+1}, where $r_k = b - Ax_k$, is minimized, or in other words,

$$\langle r_k - \alpha A r_k, Ar_k \rangle = 0, \tag{1.18}$$

which, by linearity of the inner product, yields

$$\alpha_k = \frac{\langle r_k, Ar_k \rangle}{\langle Ar_k, Ar_k \rangle}. \tag{1.19}$$

This update equation and the rule to calculate a_k yield Algorithm 2. The convergence of this algorithm is guaranteed when the matrix A is positive definite, that is when

$$\langle (A + A^*)z, z \rangle > 0 \tag{1.20}$$

for all z non-zero such that Az is defined.

Algorithm 2 Minimized Residual (MR) method [31] to solve the linear system $Ax = b$.

Require: x_0
  $r_0 = b - Ax_0$
  for $k = 0, 1, 2, ...$ **do**
    $\alpha_k = \frac{\langle r_k, Ar_k \rangle}{\langle Ar_k, Ar_k \rangle}$
    $x_{k+1} = x_k + \alpha_k r_k$
    $r_{k+1} = r_k - \alpha_k Ar_k$
  end for

1.3.3 GMRES

The Generalized Minimum Residual method (GMRES) [31], [32] is a Krylov subspace iterative method for the solution of general nonsingular systems. Krylov subspace methods are based on projections on m-th order Krylov subspaces, which are defined as

$$\mathcal{K}_m(A; b) = \langle b, Ab, ..., A^{m-1}b \rangle. \tag{1.21}$$

In other words, some vector b as well as the successive applications of A to it constitute an expanding basis, that of the Krylov subspace, which is used for projection purposes within the method. By way of orthogonalization of some direction with the Krylov subspace, a Krylov method may, for example, ensure that the next step direction and size minimizes some measure of the residual with respect to the next order Krylov subspace. Methods that use Krylov subspaces as projection spaces are plentiful: the conjugate gradients, MINRES, GCR, GMRES, etc. An extensive resource on iterative methods based on Krylov subspaces can be found in [31]. The GMRES method is a popular choice due to its ability to converge on a solution for any nonsingular matrix A, and also for its lower iteration and memory cost compared to other general Krylov methods, like GCR.

Algorithm 3 GMRES(m) method [31], [32] to solve the linear system $Ax = b$.

Require: x_0
  $r_0 = b - Ax_0$, $\beta = ||r_0||$ and $v_1 = \frac{r_0}{\beta}$.
  for $k = 1, 2, ..., m$ **do**
    $w_k^0 = Av_k$
    for $i = 1, 2, ..., k$ **do**
      $h_{i,k} = \langle w_k^{i-1}, v_i \rangle$
      $w_k^i = w_k^{i-1} - h_{i,k} v_i$
    end for
    $w_k = w_k^k$
    $h_{k+1,k} = ||w_k||$
    $v_{k+1} = \frac{w_k}{||w_k||}$
  end for
  Define H_m such that $[H_m]_{i,j}$ is $h_{i,j}$ if it is defined, $[H_m]_{i,j}$ is 0 otherwise.
  Find y_m the minimizer of $||\beta e_1 - H_m y||$
  $x_m = x_0 + \sum_{i=1}^{m} v_i y_i$

The GMRES method in particular [31] can be described in two phases. The first phase consists of the construction of an orthonormal basis from successive applications of A to the initial residual r_0, all the while recording the projection of each new direction Av_k into the previous basis $\langle Av_1, Av_2, ..., Av_{k-1}\rangle$ in the form of the columns of a *Hessenberg matrix* H_m. The second phase consists of finding some vector y_m such that it minimizes the residual $||\beta e_1 - H_m y||$, where $\beta = ||r_0||$ and e_1 is a unit vector along the first dimension. Then, the solution is recovered using $x_m = x_0 + \sum_{i=1}^{m} v_i y_i$ where y_i is the i-th component of y. In practice, the second phase can be reduced to solving a triangular linear system through appropriate Givens rotations. The reader is referred to the extensive explanations in [31] or [14] for the practical implementation of the computation of y_m. The GMRES(m) method, where m designates the amount of iterations performed per cycle, is shown in Algorithm 3.

The set of m iterations of the GMRES(m) algorithm is called a cycle. The computational cost of a cycle grows in $O(m^2)$. In practice, a tradeoff therefore has to be made regarding the size m of the GMRES cycle.

1.3.4 Preconditioning

The system operator A resulting from the discretization of the problem is often ill-conditioned [34], leading iterative algorithms to have slow convergence and the solution being more prone to numerical instabilities. The conditioning of the system can be improved with preconditioning [31], that is, by introducing an application M to create a system with the same solution, but with a better conditioned system matrix. The preconditioning can be on the left, by applying M^{-1} to both sides of (1.10), or on the right, with $M^{-1}M$ being inserted between A and u, such that the system becomes

$$AM^{-1}y = b \quad \text{with} \quad u = M^{-1}y. \tag{1.22}$$

In FWI, right-side preconditioning is often preferred due to better stability and the residual being unchanged [34].

Domain decomposition

For scalability reasons, the domain is decomposed into S overlapping subdomains Ω_j [10], and we define V^{h_j} the space of functions derived from the generating basis of V^h by keeping the trial functions associated with nodes in Ω_j. Accordingly, the waveform field u is decomposed into S partitions, which are indexed by u_j, all belonging to V^{h_j}. The subdomain Ω_j is further divided into two regions: the *owned* region – consisting of regions exclusive to the subdomain as well as owned parts of the overlapping regions – and the *ghost* region – consisting of regions of the overlap which are not owned.

The subdomain restriction operator $R_j : V^h \to V^{h_j}$ is the operator that projects a global field V^h onto its subdomain local basis V^{h_j}, such that $u_j(x) = R_j(u)(x)$. The subdomain weighting operator $D_j : V^{h_j} \to V^{h_j}$ is a diagonal operator such that $D_j(u_j)(x) = d_j(x)u_j(x)$ where d_j is in V^{h_j} and $d_j(x)$ form a partition of unity on all j. Typically, d_j is chosen to take the value of zero on ghost regions, such that the purpose of D_j is to act as a mask, zeroing out contributions on the ghost regions after any operator of type $V^{h_j} \to V^{h_j}$ is applied. Given R_j, D_j, and the field u_j on

all subdomains, the field u may be reconstructed using the adjoint operator of R_j, with

$$u = \sum_{j=1}^{S} R_j^* D_j u_j. \tag{1.23}$$

Optimized restricted additive Schwarz (ORAS) preconditioning is an effective method to build preconditioners in the context of solving the Helmholtz problem with domain decomposition. The ORAS preconditioner is built as

$$M^{-1} = \sum_{i=1}^{S} R_i^* D_i B_i^{-1} R_i, \tag{1.24}$$

where B_i is the discretization of the physics of the problem on the subdomain, including transmission boundary conditions. The application of B_i^{-1} is performed using direct methods; due to the reduction in dimension of the subdomain-local problem, doing so can be carried out at lower computational cost and parallelized on all the subdomains. Applying M^{-1} to a field y in V^h on subdomain j consists of performing the following

$$u_j = D_j B_j^{-1} y_j + R_j \sum_{i \in O(j)} R_i^* D_i B_i^{-1} y_i, \tag{1.25}$$

where $O(j)$ is the set of subdomains neighboring j, y_j is such that $y_j = R_j y$. In other words, the work carried out by the computing unit on each subdomain when applying the preconditioner consists of (a) solving the subdomain-local system by taking into account ghost regions pertaining to overlaps not owned by the subdomain, and (b) accumulating the results of (a) of neighboring subdomains on the overlaps.

Since preconditioning and domain decomposition are practical considerations for the resolution of the forward system, the rest of this chapter shall use the notation (1.10) to refer to the equation solved by the forward solver, regardless of the use of domain decomposition and preconditioner.

1.4 The inverse Helmholtz problem

The waveform solution u of the particularized Helmholtz problem (1.8) is obtained for a given angular frequency ω and squared slowness field m. Consider now that the value of the waveform d is measured on a subdomain Γ of Ω. The subsequent inverse Helmholtz problem consists of treating m as a degree of freedom and to find m which is such that u approximates d on Γ. The closeness of u to d on Γ is typically formulated as a misfit functional, and as a consequence, the process of inversion consists of a minimization process on that functional.

The inverse problem associated with the direct problem (1.8) and the knowledge of the measured waveform on Γ consists of finding m such that

$$\arg\min_m J(u) \quad \text{with } u \text{ in } V \text{ such that (1.8) holds for all } v \text{ in } V. \tag{1.26}$$

where J is a misfit functional between u and d on Γ. Figure 1.2 summarizes the domain on which the inverse problem is defined.

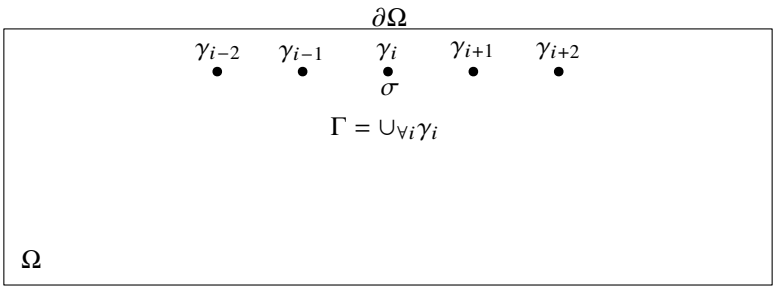


Figure 1.2: Domain on which the inverse problem 1.26 is defined. $\partial\Omega$ is the transmission boundary, σ is the emitter location, Γ is the subdomain where the waveform d is observed, which is constituted of a discrete set of points γ_i.

1.4.1 Discretization

The inverse problem is discretized to obtain an optimization problem which is solvable numerically. The squared wave slowness shall also be discretized into a finite set of degrees of freedom for the optimization to be tractable. The squared slowness field is projected onto a finite-dimensional subspace W^h, generated by a basis of shape functions not necessarily equal to V^h

$$W^h = \langle \varphi_1, \varphi_2, ..., \varphi_M \rangle, \tag{1.27}$$

where the shape functions φ_i are defined such that $\varphi_i(x_i) = 1$ for all i and $\varphi_i(x_j) = 0$ for all (i, j) with $i \neq j$, and such that they have a local support on the elements adjacent to the node. The distinction is made between W^h and V^h in that nodes and elements may be defined differently from the nodes and elements that induce the basis of V^h, *e.g.* the meshes may have a different topology and order. The projected wave slowness field m is expressed in W^h as

$$m(x) = \sum_{i=1}^{M} m_i \varphi_i(x). \tag{1.28}$$

By restricting m to W^h, the subspace of squared wave slowness fields that may be employed in the discretization of the Helmholtz problem, and adding in the explicit dependency on m to A, solving the Helmholtz problem amounts to finding u in V^h such that

$$A(m)u = b. \tag{1.29}$$

The discretized inverse Helmholtz problem consists of finding some m in W^h such that

$$\arg\min_m J(u) \quad \text{with } u \text{ in } V^h \text{ such that } A(m)u = b. \tag{1.30}$$

1.4.2 Misfit functional

The misfit functional J which is to be minimized for the resolution of the inverse problem may be defined in a variety of ways. A natural choice [2], [34] for the misfit is to define it as half the square of some norm of the error between u and d on Γ. Let V^Γ be the function space in which d lies, let

$P_\Gamma : V^h \to V^\Gamma$ be the projection operator of the waveform space V^h into V^Γ, the misfit functional typically writes

$$J(u) = \frac{1}{2}\langle P_\Gamma u - d, P_\Gamma u - d\rangle. \tag{1.31}$$

In practice, the domain Γ consists of a discrete set of R points γ_i along which point receivers are placed. In such a case, V^Γ is generated by the basis

$$V^\Gamma = \langle \delta_{\gamma_1}, \delta_{\gamma_2}, ..., \delta_{\gamma_R}\rangle, \tag{1.32}$$

and the application of P_Γ onto u writes $P_\Gamma u = \sum_{i=1}^R \delta_{\gamma_i} u(\gamma_i)$. Assuming the points γ_i are all distinct, then $P_\Gamma u$ is isomorphic to a vector $[P_\Gamma u] \in \mathbb{C}^R$ such that $[P_\Gamma u]_i = u(\gamma_i)$, and the inner product defining the misfit (1.31) is equivalent to the Euclidean dot product on $\mathbb{C}^R$,

$$J(u) = \frac{1}{2}\,[P_\Gamma u - d] \cdot [P_\Gamma u - d] = \frac{1}{2}\sum_{i=1}^{R} \overline{[P_\Gamma u - d]_i}\,[P_\Gamma u - d]_i\,. \tag{1.33}$$

1.4.3 Multi-source inverse problem

Because the number of unknowns generally exceeds the available information, inverse problems are largely ill-posed problems. A misfit functional such as (1.31) is subject to several local minima, which may not necessarily reflect a realistic distribution of the parameters. Attempts can be made at increasing the well-posedness of the inverse problem by considering a set of response observations induced by multiple source terms, which may emit waves at different frequencies and from different locations.

Consider several different instances of the direct Helmholtz problems associated with the inverse problem (1.26) acting on the same domain Ω, but with potentially different frequencies ω_i, source positions σ_i, and observation subdomains Γ_i. Once discretized, each direct Helmholtz problem consists of finding u in V^h such that

$$f_i(m, u) = A_i(m)u - b_i = 0. \tag{1.34}$$

A strategy to increase the well-posedness of the inversion process consists of defining a global misfit functional J as

$$J = \sum_{i=1}^{I} J_i(u_i) \quad \text{with } u_i \text{ such that } f_i(m, u_i) = A_i(m)u_i - b_i = 0, \tag{1.35}$$

where I is the number of considered instances of the direct Helmholtz problem and i is the index of that instance. The inverse Helmholtz problem associated with different instances of the direct problem consists of finding some m in W^h such that

$$\arg\min_m \sum_{i=1}^{I} J_i(u_i) \quad \text{with } u_i \text{ such that } f_i(m, u_i) = A_i(m)u_i - b_i = 0. \tag{1.36}$$

A few practical observations can be drawn from the direct Helmholtz problems (1.34). First, if A_i is the same for all instances of the direct problem, then solving the direct Helmholtz problems

amounts to solving a linear system with multiple right-hand sides. Second, since the global misfit functional J is a linear combination of the misfit functionals of the different instances of the problem, the derivatives of J necessary for the optimization algorithms can be calculated by superposing the corresponding derivatives of every J_i individually. In the reference solver used here, inversion is always performed for a single frequency at a time. A_i is therefore the same matrix for all i; the resolution of the linear systems involving A_i may be formulated as a multiple right-hand side system.

For the sake of clarity, the remainder of this chapter will consider the inverse problem (1.30), without loss of generality. Since the global misfit functional depends linearly on the misfit functionals of the instances of the problem, cases with multiple set of observations may be taken into account by summing the derivatives of the misfit functionals before performing the optimization.

1.5 Optimization

Solving an inverse problem of the type of (1.30) requires the minimization of the misfit functional J, which is a function of u, subject to the constraint $f(m, u) = 0$. We denote $L : W^h \to \mathbb{R}$ the *loss* functional with an explicit dependency on m such that

$$L(m) = J(u) \text{ with } u \text{ in } V^h \text{ such that } A(m)u = b, \tag{1.37}$$

where the difference between J and L is that the former evaluates the misfit for an arbitrary u in V^h, whereas the latter evaluates the misfit for u in V^h which is solution to the direct problem using m as a parameter field. This definition allows the differentiation in m of the misfit.

1.5.1 Local misfit modelling

The misfit of the functional can be expressed around a position m using a Taylor expansion,

$$L(m + \delta m) = L(m) + D_m L(m)(\delta m) + \frac{1}{2} D_m^2 L(m)(\delta m, \delta m) + O(\delta m^3), \tag{1.38}$$

where $D_x^p : (A \to B) \to (A \to A^p \to B)$ is the p-th order Gâteaux differentiation operator along variable x of the mapping on which D_x^p is applied. Since L is a continuous functional $W^h \to \mathbb{R}$, and assuming that it is first- and second-order Gâteaux differentiable along m, then by the Riesz representation theorem, there exist the gradient kernel $\nabla_m L : W^h \to W^h$ and the Hessian operator $\nabla_m^2 L : W^h \to W^h \to W^h$ such that the expansion is equivalently rewritten using inner products

$$L(m + \delta m) = L(m) + \langle \nabla_m L(m), \delta m \rangle + \frac{1}{2} \langle \nabla_m^2 L(m) \delta m, \delta m \rangle + O(\delta m^3). \tag{1.39}$$

Optimization methods [24] consist of creating a *local* linear or quadratic model $\tilde{L}$ of the misfit around m using lower order derivative terms of the Taylor expansion of L. The model allows finding a search direction p which reduces the local model. Since the real misfit is, in general, of higher order than the chosen approximating local model $\tilde{L}$, the latter and the actual misfit L are similar only near m. Therefore, taking a step along p is only appropriate for optimization if the step

is appropriately scaled. Once a proper descent direction and scale α have been found, the iterative procedure for minimization is to update an iterate n of the optimization variable, and to update it with the rule

$$m_{n+1} = m_n + \alpha p. \tag{1.40}$$

1.5.2 Derivatives calculation

As u in the misfit definition is defined under constraint, the gradient is here obtained using the adjoint method as described by [8]. The derivative of the loss L with respect to m can be evaluated as

$$\begin{aligned} D_m L(m)(\delta m) &= \partial_m L(m)(\delta m) + \partial_u L(m)(D_m u(m)(\delta m)), &(1.41)\\ &= \operatorname{Re}\left\{\langle P_\Gamma D_m u(m)(\delta m), P_\Gamma u - d\rangle\right\}. &(1.42)\end{aligned}$$

The derivative of u with respect to m can be obtained implicitly from the derivative of the constraint $f(m,u) = 0$. Indeed, if the constraint $f(m,u) = 0$ holds, then the constraint

$$\begin{aligned} D_m f(m,u)(\delta m) &= \partial_m f(m,u)(\delta m) + \partial_u f(m,u)(D_m u(m)(\delta m)), &(1.43)\\ &= \omega^2 U\delta m + A(m)\left(D_m u(m)(\delta m)\right) = 0, &(1.44)\end{aligned}$$

holds, where U is the diagonal operator such that $(Um)(x) = u(x)m(x)$. Given that $f(m,u) = A(m)u - b$ and that $\partial_m A(m)(\delta m) = \omega^2 U\delta m$, the derivative of u is given by

$$D_m u(m)(\delta m) = -\omega^2 [A(m)]^{-1} U\delta m, \tag{1.45}$$

where $[A(m)]^{-1}$ is the inverse of the operator $A(m)$. Injecting (1.45) into (1.41) yields the expression of the derivative

$$\begin{aligned} D_m L(m)(\delta m) &= -\omega^2 \operatorname{Re}\left\{\langle P_\Gamma [A(m)]^{-1} U\delta m, P_\Gamma u - d\rangle\right\}, &(1.46)\\ &= -\omega^2 \operatorname{Re}\left\{\langle U\delta m, [A(m)]^{-*} P_\Gamma^*(P_\Gamma u - d)\rangle\right\}. &(1.47)\end{aligned}$$

Introducing λ the adjoint state such that

$$[A(m)]^*\lambda = P_\Gamma^*\left(P_\Gamma u - d\right) \tag{1.48}$$

finally yields

$$D_m L(m)(\delta m) = -\omega^2 \operatorname{Re}\left\{\langle U\delta m, \lambda\rangle\right\}. \tag{1.49}$$

Calculating the gradient of a functional defined under constraint by solving two adjoint systems is called the adjoint method.

Hessian calculation

The Hessian operator $\nabla_m^2 L(m)$ is calculated using the second Gâteaux derivative $D_m^2 L$ of L with respect to m, which yields

$$\begin{aligned} D_m^2 L(m)(\delta m, \delta m) &= -\omega^2 \langle P_\Gamma D_m u(m)(\delta m), P_\Gamma [A(m)]^{-1} U\delta m\rangle\\ &\quad - \operatorname{Re}\left\{\omega^4 \langle P_\Gamma D_m u(m)(\delta m), P_\Gamma [A(m)]^{-2}(U-U)\delta m\rangle\right\}, &(1.50)\\ &= \omega^4 \langle P_\Gamma [A(m)]^{-1} U\delta m, P_\Gamma [A(m)]^{-1} U\delta m\rangle. &(1.51)\end{aligned}$$

The Hessian operator is the operator $\nabla_m^2 L(m)$ such that

$$D_m^2 L(m)(u,v) = \langle \nabla_m^2 L(m)u, v\rangle. \tag{1.52}$$

1.5.3 Line search

Although some optimization algorithms can, under some assumptions, provide a direction and scale which are sufficient for global convergence, it is not the case in general. If the optimization method uses a local model which is far from the real one, accepting the default scale yielded by the optimization method might not be sufficient to ensure the global convergence of the scheme. This may happen if the misfit has strong non-linear and non-quadratic components and that first- and second-order methods are employed. In addition to convergence problems, the stability and correctness of the optimization algorithm may *require* that some specific conditions are met on the step to be taken – this is for example the case of some quasi-Newton methods. The purpose of a line search procedure is to find an appropriate step scale from a given search direction returned by some other method, such that it ensures desired properties for global convergence and correctness of the scheme.

Let p and α_0 be a descent direction and its trial scale, returned by some optimization method. Granted that the step direction is adequate, a sufficient condition for global convergence is the respect of the Armijo-Goldstein condition, also called the sufficient decrease condition [24],

$$L(m + \alpha p) \leq L(m) + c_1 D_m L(m)(\alpha p), \tag{1.53}$$

where c_1 is in $[0, 1]$. This condition can be interpreted as an upper bound on the scale α. It is easy to see that there always exists a value of α for which this condition is met, by considering values of α that tend towards zero. A simple algorithm to find a scale α which satisfies this criterion is the Armijo backtracking line search, written in Algorithm 4, where t is the backtracking parameter.

Algorithm 4 Armijo's backtracking line search.

Require: p, α_0 and $c_1, t \in [0, 1]$
  $\alpha \leftarrow \alpha_0$
  while $L(m + \alpha p) > L(m) + c_1 D_m L(\alpha p)$ **do**
    $\alpha \leftarrow t\alpha$
  end while
  Step is αp

Besides the Armijo-Goldstein condition, another important condition is the curvature condition [24],

$$-D_m L(m + \alpha p)(p) \leq -c_2 D_m L(m)(p), \tag{1.54}$$

where c_2 is in $[0, 1]$. This condition ensures that the negative slope is reduced sufficiently after the step is taken. In addition to providing a meaningful lower bound on the scale of the step, the respect of this condition is necessary for some optimization schemes to be be stable; this is the case of quasi-Newton methods that form positive definite approximations of the Hessian, such as L-BFGS. The combination of the Armijo-Goldstein condition (1.53) and of the curvature condition (1.54) is called the Wolfe conditions.

1.5.4 First order descent

Gradient descent [24] is an optimization method based around a linear local model of the misfit,

$$\tilde{L}(m+p) = L(m) + \langle \nabla_m L(m), p \rangle. \tag{1.55}$$

The algorithm consists of choosing the descent direction to be $p = -\nabla_m L(m)$. Indeed, by choosing p as such, it is trivial to show that the local model is reduced after a step of any length s along p

$$\tilde{L}\left(m - s\nabla_m L(m)\right) = \tilde{L}(m) - s\langle \nabla_m L(m), \nabla_m L(m) \rangle \leq \tilde{L}(m), \tag{1.56}$$

for all $s > 0$. A drawback of gradient descent is that the absence of second-order information prevents the identification of any trial scale $\alpha_0 = s$ for the descent direction, and it therefore has to be tuned by hand or via some heuristics during optimization. A problem with the approach of using a fixed-step length is that in case the misfit is locally ill-conditioned, *i.e.* the wells have characteristic lengths that greatly depend on the direction, the step length will be bounded by the shortest direction. The globalization method, such as a line search, ensures that the step size respects conditions necessary for global convergence.

1.5.5 Second order descent

Second order methods, or *Newton methods*, stem from a second order local model of the misfit,

$$\tilde{L}(m+p) = L(m) + \langle \nabla_m L(m), p \rangle + \frac{1}{2}\langle \nabla_m^2 L(m) p, p \rangle. \tag{1.57}$$

This local model of the misfit is minimized when the derivative with respect to p is zero, that is when p is such that

$$\nabla_m^2 L(m) p = -\nabla_m L(m). \tag{1.58}$$

Equation (1.58) is solved for p to find the local minimum of the local misfit model, which becomes the scaled search direction which is to be used in the globalization scheme such as the line search.

In many contexts, solving (1.58) using the full form of the Hessian operator $\nabla_m^2 L(m)$ is prohibitive in complexity both in space and time, since its practical realization is a dense matrix with a number of elements growing quadratically with the number of unknowns. In addition, its calculation requires a full inversion of the system operator A. Instead, two formulations of the method which do not require the full knowledge of the Hessian are employed: quasi-Newton and iterative methods. Quasi-Newton methods construct an approximation of the application of the inverse Hessian, based on the knowledge of the gradients at previous iterations. The iterative methods consist of evaluating the application of the Hessian using finite differences, and to perform iterations of an iterative system resolution algorithm on the equation (1.58).

Iterative methods

Using iterative methods [24], a descent direction p is found by using an iterative linear system solution algorithm on the residual of (1.58). One advantage of iterative methods is that the full knowledge of the linear operator $\nabla_m^2 L(m)$ is not required, and that it suffices to be able to evaluate

the application of the Hessian operator to p, *i.e.* $\nabla_m^2 L(m)p$. The second order truncation of the Taylor expansion of the gradient, which writes

$$\nabla_m L(m+p) = \nabla_m L(m) + \nabla_m^2 L(m)p + O(p^2), \tag{1.59}$$

is used to create a finite difference scheme for the evaluation of the forward Hessian application by neglecting the higher order terms of (1.59)

$$\nabla_m^2 L(m)p \simeq \nabla_m L(m+p) - \nabla_m L(m). \tag{1.60}$$

In (1.60), the equality holds when the Hessian operator does not depend on m, this is the case when the misfit $L(m)$ is linear or quadratic. Using the approximation (1.60) of the Hessian application, the residual r of (1.58) becomes

$$r(p) = -\nabla_m L(m+p). \tag{1.61}$$

The exact or approximate Hessian application (1.60) is used in an iterative system solver to find either the value of p solution of (1.58), or an approximation of it. When a limited amount of iterations are performed to find only an approximation of p, this methodology is referred to as the *truncated Newton method*.

Without regularization, the Hessian operator is not of full rank, therefore most iterative algorithms adequate for singular systems may be employed, such as the Richardson iteration, the minimized residual (the latter two described in Section 1.3), and the steepest descent methods, depending on the assumptions on the Hessian operator. Interestingly, the Richardson method reduces to the fixed-step gradient descent, while the steepest descent and minimized residual methods correspond to automatically scaled gradient descent. However, by construction, the Hessian operator is symmetric, if it is also positive-definite, the conjugate gradient algorithm is naturally well-suited to iteratively solve (1.58). The conjugate gradient algorithm [31] is a particular case of Krylov methods that uses an assumption of symmetric positive definiteness (SPD) of the system operator to derive a procedure that does not need to store a basis; the procedure is shown in Algorithm 5.

Algorithm 5 Conjugate gradient algorithm [31], as used for solving the Newton equation.

Require: p_0
  $r_0 = -\nabla_m L(m + p_0)$
  $s_0 = r_0$
  for $k = 0, 1, ..., c$ **do**
    $\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle \nabla_m^2 L(m) s_k, s_k \rangle}$
    $p_{k+1} = p_k + \alpha_k s_k$
    $r_{k+1} = r_k - \alpha_k \nabla_m^2 L(m) s_k$
    $\beta_k = \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle}$
    $s_{k+1} = r_{k+1} + \beta_k s_k$
  end for
  $p = p_c$

L-BFGS (quasi-Newton)

L-BFGS is a quasi-Newton method [2] that finds a search direction and its scale using an approximation of the inverse Hessian built using gradient differences from the past c iterations. In other

words, an iteration of the method calculates the step direction p as

$$p = -B_n^{-1} \nabla_m L(m), \tag{1.62}$$

where B_n^{-1} is the inverse Hessian approximation at step n. Algorithm 6 shows the application of the L-BFGS approximation of the inverse Hessian application to the gradient.

Algorithm 6 L-BFGS inverse Hessian application $-B_n^{-1} \nabla_m L(m)$ for iteration $n > 0$.

$q \leftarrow \nabla_m L(m)$
for $i = n-1, n-2, ..., n-c$ **do**
    $\alpha_i = \rho_i \langle s_i, q \rangle$
    $q \leftarrow q - \alpha_i y_i$
end for
$z \leftarrow \frac{\langle s_{n-c}, y_{n-c} \rangle}{\langle y_{n-c}, y_{n-c} \rangle} q$
for $i = n-c, n-c+1, ..., n-1$ **do**
    $\beta_i = \rho_i \langle y_i, z \rangle$
    $z \leftarrow z + (\alpha_i - \beta_i) s_i$
end for
$\rho_n = \langle y_n, s_n \rangle^{-1}$
$p = -z$
Find an appropriate α and set $\Delta m = \alpha p$
$s_n = \Delta m$
$y_n = \nabla_m L(m + \Delta m) - \nabla_m L(m)$
$m \leftarrow m + \Delta m$

For the L-BFGS Hessian approximation to be positive definite and thus for p to be a descent direction, it is necessary that $\langle s_n, y_n \rangle > 0$ for all steps n of the descent, which is the case when the step size complies to the Wolfe conditions. The line search employed to find an appropriate α in Algorithm 6 should therefore enforce those conditions.

1.5.6 Gauss-Newton

The Gauss-Newton method is an optimization algorithm [24] fit to optimize functionals defined as a sum of squares of residuals, that is, functionals that may be written as

$$J(\beta) = \frac{1}{2} \langle r(\beta), r(\beta) \rangle, \tag{1.63}$$

where r is some kind of error vector. In the context of FWI, and of functionals defined as (1.31), the Gauss-Newton method (with $r = P_\Gamma u - d$) is amenable to be used as a Newton method.

The Gauss-Newton method consists of minimizing a misfit that has been made quadratic in m through the linearization of the residual vectors $P_\Gamma u - d$ in m around the background field $\tilde{m}$.

Noting $\tilde{u}$ the field such that $f(\tilde{m}, \tilde{u}) = 0$, the misfit arising from the linearized residuals $\tilde{L}$ writes

$$\tilde{L}(\delta m) = \frac{1}{2}\langle r(\tilde{m}) + D_m r(\tilde{m})(\delta m), r(\tilde{m}) + D_m r(\tilde{m})(\delta m)\rangle \tag{1.64}$$

$$= \frac{1}{2}\langle r(\tilde{m}), r(\tilde{m})\rangle + \mathrm{Re}\left\{\langle D_m r(\tilde{m})(\delta m), r(\tilde{m})\rangle\right\} + \frac{1}{2}\langle D_m r(\tilde{m})(\delta m), D_m r(\tilde{m})(\delta m)\rangle \tag{1.65}$$

$D_m r(\tilde{m})$ being equal to $P_\Gamma D_m u(\tilde{m})$, the misfit becomes

$$\begin{aligned} &= L(\tilde{m}) - \omega^2 \,\mathrm{Re}\left\{\langle P_\Gamma [A(\tilde{m})]^{-1}\tilde{U}\delta m, P_\Gamma \tilde{u} - d\rangle\right\} \\ &\quad + \frac{\omega^4}{2}\langle P_\Gamma [A(\tilde{m})]^{-1}\tilde{U}\delta m, P_\Gamma [A(\tilde{m})]^{-1}\tilde{U}\delta m\rangle, \end{aligned} \tag{1.66}$$

where $\tilde{U}$ is the diagonal operator such that $(\tilde{U}m)(x) = \tilde{u}(x)m(x)$.

The step direction and scale p to be used in the line search is given by

$$p = \arg\min_{\delta m} \tilde{L}(\delta m). \tag{1.67}$$

In other words, the Gauss-Newton method consists of minimizing a non-quadratic least squares misfit $L(\tilde{m} + \delta m)$ by minimizing a succession of quadratic misfits $\tilde{L}(\delta m)$. $\tilde{L}(\delta m)$ is optimized by differentiating along δm and finding δm such that

$$\omega^2 \left[P_\Gamma [A(\tilde{m})]^{-1}\tilde{U}\right]^* P_\Gamma [A(\tilde{m})]^{-1}\tilde{U}\delta m = \mathrm{Re}\left\{\left[P_\Gamma [A(\tilde{m})]^{-1}\tilde{U}\right]^* [P_\Gamma \tilde{u} - d]\right\}. \tag{1.68}$$

This equation may be solved using the conjugate gradients, as explained in Section 1.5.5.

1.5.7 Linearization

Another way to derive the Gauss-Newton equation consists of linearizing the Helmholtz problem. The inverse Helmholtz problem can be linearized by considering that m and u are a superposition of a background field and a perturbation field,

$$m = \tilde{m} + \delta m \quad \text{and} \quad u = \tilde{u} + \delta u, \tag{1.69}$$

and by considering that any product of perturbation fields is negligible. The constraint can be developed

$$f(m, u) = A(\tilde{m} + \delta m)(\tilde{u} + \delta u) - b, \tag{1.70}$$

and the expression of the operator A being linear in m, its value can be *exactly* expressed with a first order Taylor expansion

$$= [A(\tilde{m}) + \partial_m A(\tilde{m})(\delta m)]\,(\tilde{u} + \delta u) - b. \tag{1.71}$$

Given that $\partial_m A(\tilde{m})(\delta m)$ is $\omega^2 \delta m$, and that the constraint f must hold for the background field such that $f(\tilde{m}, \tilde{u}) = 0$, the constraint becomes

$$= A(\tilde{m})\delta u + \omega^2 \tilde{U}\delta m + \omega^2 \delta U \delta m. \tag{1.72}$$

Finally, by neglecting the product of perturbed fields $\omega^2 \delta U \delta m$ in the last term, a linearized constraint δf can be obtained

$$\delta f(\delta m, \delta u) = A(\tilde{m})\delta u + \omega^2 \tilde{U}\delta m. \tag{1.73}$$

In the context of the Helmholtz problem, this linearization process is called the Born approximation [37].

The linearized inverse problem misfit $\tilde{J}$ becomes a function of the perturbed waveform field δu, that is

$$\tilde{J}(\delta u) = \frac{1}{2}\langle P_\Gamma(\tilde{u} + \delta u) - d, P_\Gamma(\tilde{u} + \delta u) - d\rangle. \tag{1.74}$$

The linearized inverse Helmholtz problem then consists of finding δm such that

$$\arg\min_{\delta m} \tilde{J}(\delta u) \quad \text{with } \delta u \text{ such that } \delta f(\delta m, \delta u) = A(\tilde{m})\delta u + \omega^2 \tilde{U}\delta m = 0, \tag{1.75}$$

where $\tilde{u}$ and $\tilde{m}$ are such that $f(\tilde{m}, \tilde{u}) = 0$. Substituting $\delta u = -\omega^2 [A(\tilde{m})]^{-1} \tilde{U}\delta m$ into (1.74) yields the Gauss-Newton misfit (1.66).

Chapter 2

Practical tooling and benchmark

The aim of this chapter is to present the practical framework in which our numerical experiments are realized. This chapter is divided into three main sections. The first section is dedicated to a brief presentation of the main piece of tooling used to carry out the experiments; in particular, the reference solver will be presented. Then, the reference models that are to be used throughout the experiments of this work will be introduced, alongside the process by which the initial models (*i.e.* input model to the inversion algorithm) are created from the reference model.

2.1 Tooling

The experimental work is based on several pieces of tooling and libraries. The main experimental component is a distributed FWI code developed by B. Martin [22]. Figure 2.1 presents the main dependencies and illustrates how the main tools depend on each other.

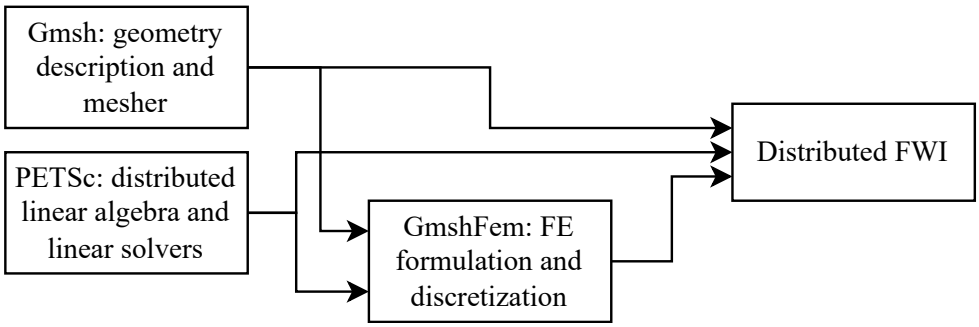


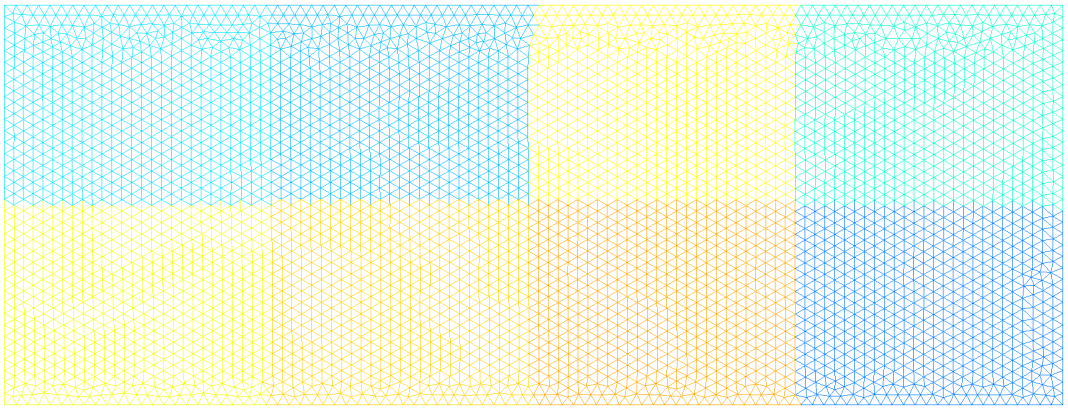
Figure 2.1: Simplified dependency tree of the Distributed FWI, the experimental code.

2.1.1 Reference solver

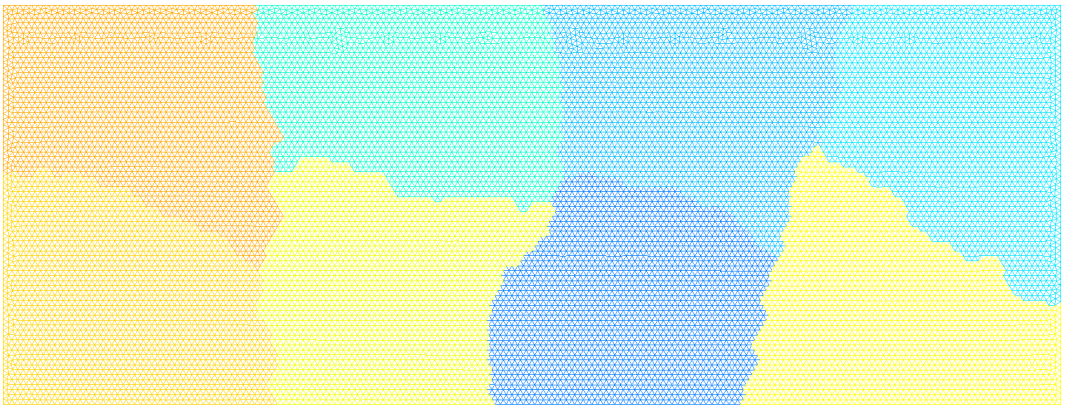
The solver extends, where applicable, the knowledge acquired by X. Adriaens during his thesis [1] on FWI, and explores avenues of improving the scalability of the FWI process through the use of domain decomposition with iterative methods. The iterative methods are made tractable by the use

of an ORAS preconditioner. The purpose of this subsection is to elaborate on the inner workings of the solver, and to describe it in the context of the techniques outlined in Chapter 1, to make the reader more familiar with the experimental framework.

Meshing and partitioning. From a file describing the reference model contents and its dimension, the program first creates a description of the domain and then meshes it using Gmsh. The layout and number of sources is a parameter of this process; to model point sources, each source is assigned an *embedded* point in the mesh, which is a way to constrain Gmsh to put a mesh node at a specified location. Once the meshing is done, the discretized domain is partitioned to accomodate for the number of subdomains, being equal to the amount of processes launched. Two partitioning modes can be chosen: a checkerboard layout and METIS [18]. The checkerboard layout consists of a simple partition described by $N \times M$, where N is the number of subdivisions along the width and M is the number of subdivisions along the height. METIS is a graph partitioning software that automatically creates partitions with good load balancing for locally refined meshes, *i.e.* the partition sizes will be mostly uniform in terms of number of elements even if the mesh is locally finer. Figures 2.2a and 2.2b illustrate two domains that have been partitioned to accomodate for



(a) Checkerboard partitioning with 4×2 subdomains.



(b) METIS partitioning with 8 subdomains.

Figure 2.2: Illustration of the partitioning schemes used by the reference solver and supported by Gmsh. In each case, the partitioner accommodates for 8 subdomains.

eight subdomains, with a checkerboard pattern and with METIS respectively.

Reference waveform data. The reference waveform data at the receivers is computed by solving the Helmholtz problem with the ground truth as velocity model. Since the focus is here on the methods, the reference data is obtained synthetically using the model itself. This process is carried out for each frequency of interest.

Initial guess. The initial model of the inversion process is obtained by smoothing the reference model using a diffusion equation, thereby removing the small scale features of the model. In the present study, the initial model is based on preliminary knowledge of the reference model, as opposed to being a true guess.

Optimization. The misfit of the inverse Helmholtz problem can be minimized in two different ways. The main way, and the focus of this thesis, is by using the Gauss-Newton method (see Chapter 1) through the linearized inverse problem. Using Gauss-Newton, the Newton equation can either be solved using projection methods, or with fixed step gradient descent. The other possibility is to solve the non-linearized inverse Helmholtz problem directly; this is done using fixed-step gradient descent or L-BFGS. When solving the full inverse Helmholtz problem directly, the Helmholtz equation operator $A(m)$ is re-discretized at each step. The globalization scheme used for steps in the full inverse Helmholtz problem (*e.g.* L-BFGS step, Gauss-Newton step) is Armijo's backtracking linesearch [24], with a backtracking coefficient of 0.5 and $c_1 = 0$.

Forward and adjoint resolution. Whether the inverse Helmholtz problem misfit is minimized directly or using Gauss-Newton through the linearized inverse problem, the forward and adjoint fields are necessary to calculate the gradient, which is in turn necessary for the optimization schemes. The forward and adjoint fields can be solved for using either direct and iterative methods. When iterative methods are employed, the system is preconditioned using ORAS; the ORAS preconditioner is built according to the subdomains defined by the mesh partitioning step. The calculation of the different terms of the sum in (1.24) is distributed among the processes. The default convergence criterion of the iterative schemes is the relative residual reducing to below 10^{-6}, unless noted otherwise.

2.1.2 PETSc

PETSc [3], [4] is a toolkit for the scalable solution of partial differential equations. At its core, PETSc offers facilities for distributed linear algebra, including distributed sparse or dense matrix-vector manipulation and system resolution facilities based on either direct or iterative methods. In the reference solver, PETSc is used as the go-to linear algebra library.

2.2 Reference cases

The aim of this section is to present the reference models that are used in the experiments. All these models are two-dimensional tomographies of the acoustic squared wave slowness field m. Since

these models have highly different spatial scales, ranging from centimeters to dozens of kilometers, the characteristic length of the finite element discretization is presented as well.

2.2.1 Marmousi case

The main model that is to be investigated in this work is the Marmousi model [35]. The Marmousi model was created in 1988 by the Institut Français du Pétrole based on the North Quenguela Trough in the Cuanza Basin in Angola, with the aim of serving as a common reference model for a workshop on velocity estimation methods. The data of the Marmousi model was generated synthetically, but with the aim of being geologically realistic and sufficiently complex, as the model contains many reflectors (transitions from low to high slowness) and steep – sometimes discontinuous – velocity gradients in all directions. To this day, the Marmousi model remains a widely recognized and used 2-D benchmark for acoustic tomography methods, including FWI [1]. The Marmousi model as used in this work is shown in Figure 2.3. A fixed layer of water, with wave velocity of $1500\,\mathrm{m\,s^{-1}}$, has been added above and is considered a known part of the model during inversion. In this thesis, experiments based on the Marmousi model use a triangular regular finite element discretization with a side length of $l_c = 0.05\,\mathrm{km}$.

2.2.2 2004 BP

Another model of interest in this study is the 2004 BP model [6]. This model was created in 2004 by BP to address the need for a new, yet unknown, model for a blind test at a velocity estimation

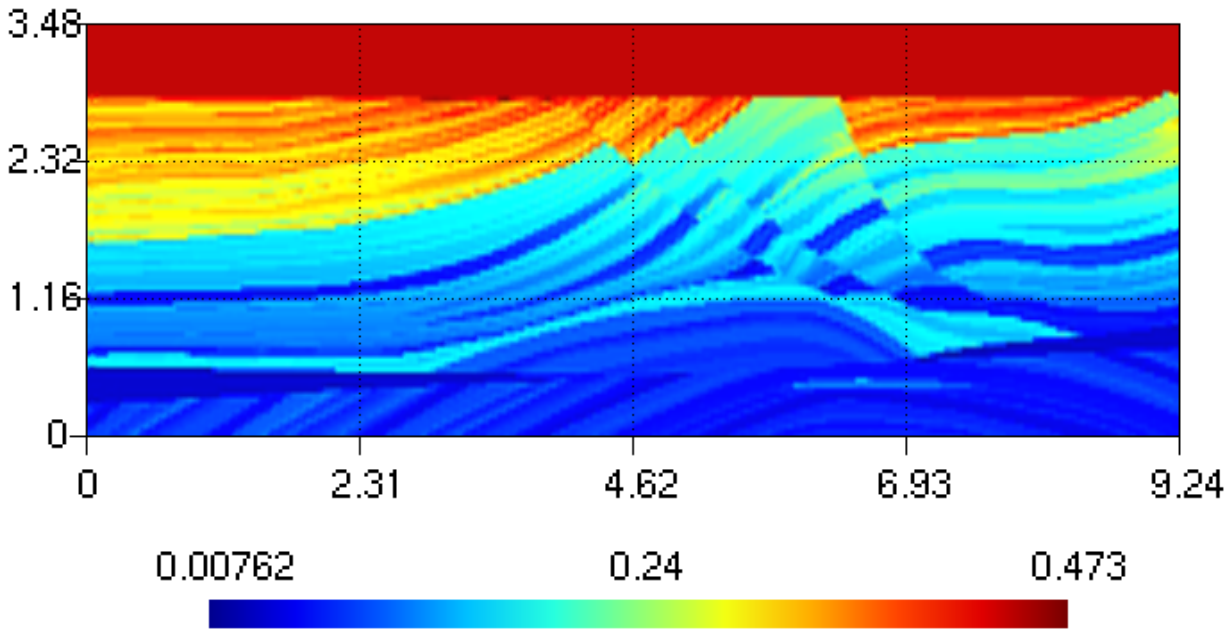


Figure 2.3: The Marmousi squared wave slowness model [35]. The distances are expressed in kilometers, and the squared wave slowness field is expressed in $\mathrm{s^2\,km^{-2}}$. A water layer has been added on top and is considered known during inversion.

workshop. Its scale is relatively large, being 67 km wide and 12 km deep. The model is also created synthetically designed to be realistic and representative of real geological features found in the Gulf of Mexico, West Africa, the Caspian sea, offshore Trinidad and the North sea. In particular, this model contains large salt bodies (some floating, some deeply rooted) as well as velocity anomalies in sub-salt layers. In addition, the model also features several small slow velocity anomalies which require higher frequencies to be properly imaged and were challenging to several tomography techniques at that time. To this day, 2004 BP remains another popular tomography benchmark model. The squared slowness field of the 2004 BP model is illustrated in Figure 2.4. The model is discretized using regular triangular finite elements. Due to the large scale of this model compared to the Marmousi case, the frequencies involved to capture the larger scale features of the model are lower and the side length of the finite element is set to a longer value of $l_c = 0.2\,\text{km}$.

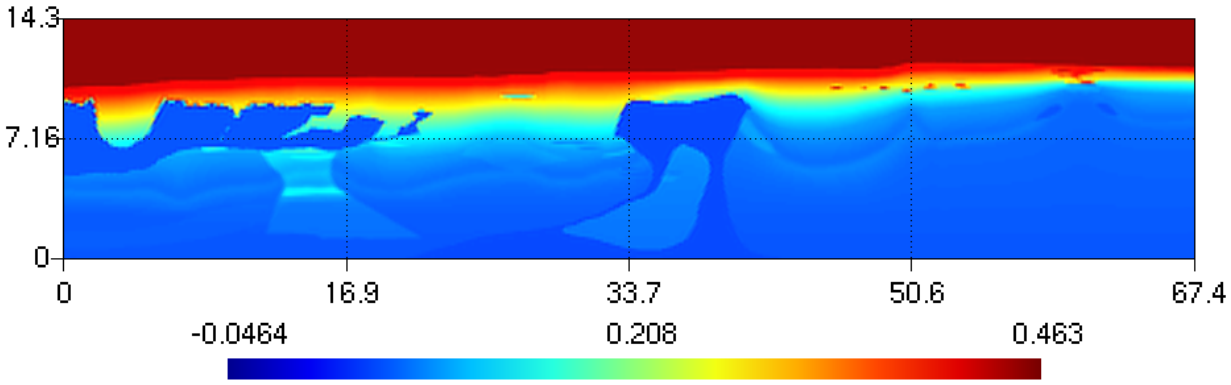


Figure 2.4: The 2004 BP squared wave slowness model [6]. The distances are expressed in kilometers, and the squared wave slowness field is expressed in $\text{s}^2\,\text{km}^{-2}$. A water layer has been added on top and is considered known during inversion. The salt deposits are the complex structures visible on the left (floating) and on the center (deeply rooted).

2.2.3 T-shaped reflectors

The last reference model used in this investigation is a synthetic near surface imaging scenario proposed by [23] and further investigated by [1]. This model consists of a side tomography of two T-shaped concrete beams embedded upside down within a surface. A reflecting layer representing a change in composition of the soil is placed underneath the beams. The wave velocity in the soil, the beams and the reflective layer is of $300\,\text{m}\,\text{s}^{-1}$, $4000\,\text{m}\,\text{s}^{-1}$ and $400\,\text{m}\,\text{s}^{-1}$ respectively. The main difficulty of this model comes from the multiple reflections that exist between the two beams, drastically worsening the conditioning of the discretized Helmholtz problem. The second difficulty comes from the high contrast between the beams and the soil, and between the soil and the reflecting layer. The scale of the problem is smaller than that of Marmousi and 2004 BP, being 30 m wide and 4.2 m high (including a fixed soil layer). Figure 2.5 illustrates the squared slowness field of this reference model. In this work, this model is discretized using regular triangular finite elements with a characteristic length of $l_c = 0.08\,\text{m}$.

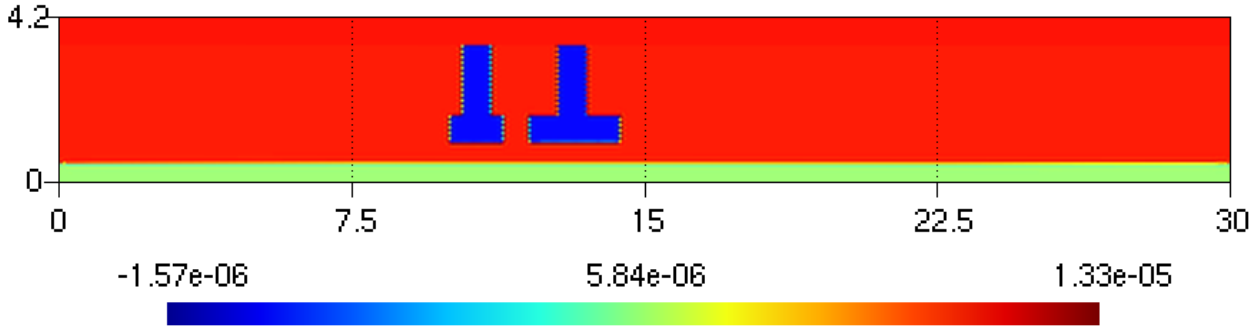


Figure 2.5: The T-shaped reflector squared wave slowness model [1], [23]. The distances are expressed in meters, and the squared wave slowness field is expressed in $s^2\,m^{-2}$. A soil layer has been added on top and is considered known during inversion. The minimum and maximum values of the squared slowness are rendered inaccurate due to undershoots and overshoots during projection.

2.2.4 Emitters and receivers handling

Numerical experience showed that placing the emitters and the receivers into the part of the domain where m is subject to optimization leads to a more difficult inversion. In addition, in a practical setting, it is likely that the wave velocity of the medium in which the receivers are placed is known. Following this observation, for all the reference models, the domain is subdivided into two regions; a region Ω_α which is the *unknown* region and is subject to optimization, and a region Ω_β which is located above the region Ω_α and whose squared slowness field is supposed to be known and fixed during optimization. In the Marmousi and 2004 BP cases, the Ω_β region is considered to be water above the soil, whereas in the T-shaped reflectors case, it is considered to be soil. The emitters and receivers are placed in the region Ω_β, such that the location of the receiver and its immediate surroundings are not on a region of the squared wave slowness subject to optimization. The depth of the Ω_β region is set to be 20% of the depth of the reference model.

In all three cases, and unless noted otherwise in the experiment description, 40 emitters/receivers are placed in a row at 50% of the depth of the Ω_β region. The first and last receiver of the row are placed at $1/52$ and $51/52$ of the model width, respectively. Figure 2.6 illustrates the interface between the two regions on the Marmousi model, as well as the layout of the emitters/receivers in the Ω_β region.

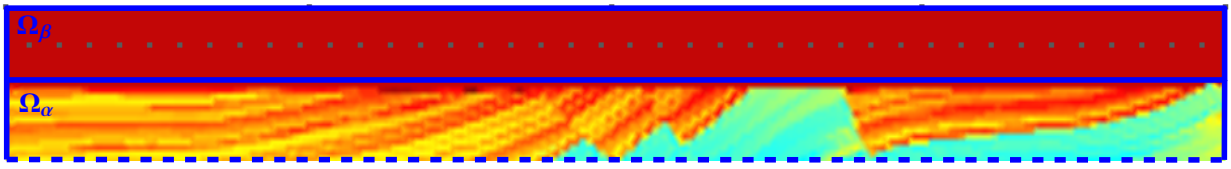


Figure 2.6: Highlight of the interface between the Ω_α and Ω_β region. The Ω_α region is subject to optimization, whereas Ω_β is fixed during optimization. A row of 40 evenly spaced emitters/receivers are placed at half the depth of the Ω_β region, starting at $1/52$ to $51/52$ of the width. The Ω_α region continues below the content of the figure.

For the inversion, all the emitters and receivers are considered of use during the inversion, and the inverse problems is treated concurrently as explained in Section 1.4.3. For each receiver, the misfit between the synthetic and observed data is calculated, and this misfit is used for the calculation of the gradient local to the source, and then the global gradient is the sum of the local gradients. Since the same frequency is considered during inversion, the forward and adjoint problems are formulated as multi-RHS linear systems.

2.2.5 Initial guess creation

The initial guesses are created by smoothing the reference model using the diffusion equation with a finite difference approximation of the time derivative. That is, the initial guess is found by finding m such that

$$\begin{cases} m(x) = m_0(x) + t\nabla_x^2 m(x) & \text{on } \Omega, \\ \partial_n m(x) = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.1}$$

where m is the initial guess of the squared slowness field and and m_0 is the reference value of the squared slowness field. This equation is solved in its weak form using GmshFem before starting the optimization. The smoothing factor t is expressed in meters squared and is dependent on the spatial scale of the problem. The value of t is set to $0.1\,\mathrm{m}^2$, $1.0\,\mathrm{m}^2$ and $5.0\,\mathrm{m}^2$ for the Marmousi, 2004 BP and T-shaped reflectors models respectively. The initial models are shown in Appendix A for each reference.

Chapter 3

One-shot methods

The inversion techniques covered in Chapter 1 all place the gradient of the misfit as the centerpiece of the optimization process. Indeed, all optimization techniques presented in Chapter 1 can be viewed as some form of preconditioned gradient descent, $p = B^{-1}\nabla_m J(m)$, where the application of the preconditioner B^{-1} only depends, at most, on gradient information at past iterates (quasi-Newton methods), or otherwise on the ability to estimate the Hessian application using only the gradient (inexact Newton methods). The derivative of the misfit is calculated using (1.49), where the waveform field u and the adjoint state field λ are obtained by solving the linear systems

$$A(m)u = b, \tag{3.1}$$

$$[A(m)]^*\lambda = P_\Gamma^*(P_\Gamma u - d). \tag{3.2}$$

In the case where the linearized inverse problem, or equivalently Gauss-Newton, is employed as an inversion strategy, the gradient is obtained from the negative residual of (1.68). The background field $\tilde{u}$ is computed using the above equations for a reference background field $\tilde{m}$. The waveform field perturbation δu and the adjoint field λ caused by a perturbation δm in the model field correspond to the solutions of the following equations,

$$\tilde{A}\delta u = -\omega^2\tilde{U}\delta m, \tag{3.3}$$

$$\tilde{A}^*\lambda = P_\Gamma^*\left[P_\Gamma\delta u - \delta d\right], \tag{3.4}$$

where $\tilde{A} = A(\tilde{m})$ and $\delta d = d - P_\Gamma\tilde{u}$.

In both the non-linearized and the linearized inversion, the evaluation of the gradient requires the solution of two linear systems. These linear systems can be solved with a direct solver or using an iterative algorithm with an efficient preconditioner, such as ORAS. For large scale systems, the direct factorization of the matrices may become prohibitive [34], and the use of iterative algorithms with domain decomposition becomes an attractive option.

When iterative algorithms are used to calculate the waveform and adjoint state field, the convergence criteria of these methods become parameters of the inversion. The FWI process can be seen as a two-layered loop; the inner loops consisting of performing the iterations of the chosen iterative algorithm to evaluate u and λ until the convergence criterion is met; the outer-layer

loop consisting of calculating the gradient using the previously calculated u and λ fields, and then performing a step in m in some gradient-related direction. The key idea of one-shot optimization [7], [37] is to *unroll* a few iterations of the inner loops, such that the outer loop becomes one of calculating u, λ and m concurrently instead of sequentially.

The purpose of this chapter is to investigate the practical usage of one-shot optimization in the context of domain decomposition FWI. The one-shot principle is applied on the iterative resolution algorithms used to calculate u and λ, and the convergence of gradient descent is mapped as a function of the algorithm's parameters. Then, variants of gradient descent integrating adaptive parameters are proposed and compared with state-of-the-art algorithms. All along, we will focus on the one-shot paradigm applied to the linearized inverse Helmholtz problem.

3.1 One-shot paradigm

In the context of the inversion of the linearized inverse Helmholtz problem where iterative algorithms are used to solve the forward and adjoint problems, let us introduce the ℓ-th step of an abstract iterative algorithm for the resolution of linear systems defined such that

$$\delta u_{\ell+1} = \langle \ell\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}\delta u = -\omega^2 \tilde{U}\delta m \rangle \tag{3.5}$$

$$\lambda_{\ell+1} = \langle \ell\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}^*\lambda = P_\Gamma^*(P_\Gamma \delta u - \delta d) \rangle \tag{3.6}$$

In practice, the actual definition of `solve` is dependent on the chosen algorithm, *e.g.* Richardson or GMRES. In what follows, we will denote by `grad` the function that calculates the gradient of the misfit as a function of δu and λ, such that the misfit gradient at the k-th iteration is noted $g = \texttt{grad}(\delta u_k, \lambda_k)$. Finally, we define the `optimize` algorithm such that

$$\delta m^{n+1} = \langle n\text{-th iterate of } \texttt{optimize} \rangle. \tag{3.7}$$

Algorithm 7 Abstract FWI procedure using an iterative algorithm for the waveform and adjoint state fields.

Require: `solve` and `optimize`
  $\delta u^0 = 0; \lambda^0 = 0$
  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

This abstract representation of algorithms involved in FWI allows for a "plug-and-play" description of each possible variant. With these definitions, the standard inversion workflow, as presented in Chapter 1 is described by Algorithm 7.

In Algorithm 7, a and b are assumed to be sufficiently high such that a stopping criterion is met for the resolution of the forward and adjoint fields. Assuming that the inner loops for the calculation for δu and λ converge, then the optimization is said to be *uncoupled* from the system resolution. Since the state of δu and λ after the inner loops is not a function of their respective states before the inner loops, the gradient g is only a function of the optimization variable δm and not of the history of values taken by δu and λ.

One-shot optimization consists of truncating the execution of the inner loops, such that δu and λ have not converged yet. By doing so, the states of δu and λ posterior to the inner loops become a function of δm, and their respective states prior to the inner loops. Since the gradient is not only a function of δm, but also of the previous states of δu and λ, the optimization and solutions of the waveform and adjoint state problems are said to be *coupled*.

3.1.1 One-step one-shot

The simplest variant of one-shot method is the one-step one-shot method [37]; here, the inner loops are truncated to only one iteration, and the optimization and the forward and adjoint resolutions are performed concurrently. The one-step one-shot optimization is shown in Algorithm 8.

Algorithm 8 Abstract FWI using one-step one-shot optimization.

Require: `solve` and `optimize`
  $\delta u^0 = 0; \lambda^0 = 0$
  for $n = 1, 2, \ldots$ **do**
    $\delta u^n = \langle n\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}\delta u = -\omega^2 \tilde{U}\delta m^n \rangle$
    $\lambda^n = \langle n\text{-th iterate of } \texttt{solve} \text{ for } \tilde{A}^*\lambda = P_\Gamma^*(P_\Gamma \delta u^{n-1} - \delta d) \rangle$
    $g^n = \texttt{grad}(\delta u^n, \lambda^n)$
    $\delta m^{n+1} = \langle n\text{-th iterate of } \texttt{optimize} \rangle$
  end for

In Algorithm 8, the values taken by δu and λ after the `solve` iterations are strongly dependent on their prior state, as well as the resolution method being used (algorithm, preconditioner quality, etc.). The inversion process strongly couples the optimization of δm with the solution of the forward and adjoint systems.

Convergence

The convergence of the one-step one-shot method has been extensively studied by Bonazzoli et al. in [7] in the case where the algorithm used for the resolution of the forward and adjoint problems is the stationary iteration, and that the optimization algorithm is gradient descent with fixed step size τ. One of their main results concerning one-step one-shot inversion is a convergence theorem

providing an upper bound on τ to guarantee convergence, which depends on the convergence rate of the stationary method.

3.1.2 Concurrent multi-step

The convergence theorem of the one-step one-shot method presented in Section 3.1.1 produces steps that may, in practical scenarios, be restrictive for efficient inversion. One may wish to improve the accuracy on δu and λ before the calculation of the gradient, such that the latter is more accurate and steps taken in its direction potentially decrease the misfit function more effectively. Bonazzoli et al. suggested a generalization of the one-step one-shot method by introducing a new parameter k corresponding to a number of times that the two `solve` iterations are executed before calculating the gradient. The resulting concurrent *multi-step* or *k-step* one-shot algorithm is shown in Algorithm 9.

Algorithm 9 Abstract FWI using concurrent k-step one-shot optimization.

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

In this work, we refer to Algorithm 9 as concurrent, because the iterations to calculate δu and λ are performed in the same loop instead of two successive loops. Due to the equation $\tilde{A}\delta u = -\omega^2 \tilde{U}\delta m$ being independent of the adjoint state, concurrency has no effect on the intermediate values taken by δu during the inner loop. Concurrency has however an effect on λ, since the right-hand side of the equation $\tilde{A}\lambda = P_\Gamma^* [P_\Gamma \delta u - \delta d]$ changes at each iteration of the inner loop.

Convergence

The convergence of the multi-step one-shot method has likewise been studied by Bonazzoli et al. [7] in the case where the minimization algorithm is gradient descent with fixed size τ and the system resolution algorithm is the stationary iteration. One of their main results is a theorem providing an upper bound on τ to guarantee convergence.

3.1.3 Sequential multi-step

While rigorous convergence proofs were obtained for Algorithms 8 and 9 by Bonazzoli et al., these algorithms suffer from a few practical disadvantages. A first disadvantage is the changing right-

hand side of the adjoint problem at each iteration of the inner loop; in particular, if the algorithm `solve` chosen for the resolution depends on some basis built from the right-hand side, such as GMRES, a right-hand side change could be at best an expensive overhead without a dedicated rework of these algorithms [26]. A second disadvantage is that in the inner loops, solver iterations on the adjoint system use intermediate values of δu on the right-hand side, rather than the final value.

In this work, we propose the investigation of a sequential multi-step one-shot algorithm. Since the set of values taken by δu along the k iterations of the inner loop is independent of λ, the loop for the calculation of the new state of δu is factored out. The adjoint-state is then calculated using the final value of δu in the right-hand side. The resulting algorithm is shown in Algorithm 10.

Algorithm 10 Abstract FWI using sequential k-step one-shot optimization.

Require: `solve` and `optimize`
  $\delta u^0 = 0; \lambda^0 = 0$
  for $n = 1, 2, \ldots$ **do**
    $\delta u_0^n = \delta u^{n-1}; \lambda_0^n = \lambda^{n-1}$
    for $\ell = 1, 2, \ldots, k$ **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 $\ell = 1, 2, \ldots, k$ **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

Algorithm 10 is qualified of sequential, in that δu is calculated first, and then λ is calculated using the final value of δu. Algorithm 10 is more reminiscent of the abstract FWI procedure outlined in Algorithm 7, except that the iterations on δu and λ are explicitly truncated after k iterations.

3.2 Fixed step one-shot

This section is dedicated to the experimental investigation of the sequential multi-step one-shot scheme shown in Algorithm 10 when the optimization algorithm `optimize` is the fixed step gradient descent, that is

$$\delta m^{n+1} = \delta m^n - \tau g^n. \tag{3.8}$$

Although in practice, some sort of adaptive step scheme is desirable to avoid wasting computational resources while ensuring global convergence, the purpose of this section is to gain some qualitative insight into the behavior of the optimization variables (the exact and estimated losses and the gradient) as the iterations of the gradient descent algorithm advance. The outcome is a set of

observations and perspectives on the practical implementation of the one-shot method that will guide the later experiments of this manuscript.

In order to limit the number of parameters to investigate, this fixed step gradient descent investigation will place itself in the specific case of the linearized inverse Helmholtz problem at some given frequency, where the reference model is the Marmousi case with 40 sources. While an analysis of several other reference cases (other reference models, other frequencies, mesh refinements, etc.) would be necessary to rigorously draw generalizable conclusions and to devise heuristics, this experiment is sufficient to introduce the general behavior of one-step one-shot in practice.

In this section, the convergence of the unscaled ORAS-preconditioned Richardson iteration will be studied to motivate the transition to automatically scaled methods such as MR and GMRES(k). Then the behavior and cost of the inversion convergence with the MR and GMRES(k) algorithms as a function of the other parameters will be presented through the use of convergence maps. Finally, the influence of the number of subdomains will be investigated for GMRES(k) as system resolution algorithms.

3.2.1 Preconditioned Richardson method

The multi-step one-shot variants presented in the previous section are applied on the ORAS-preconditioned unscaled Richardson iteration, that is,

$$x^{n+1} = x^n + M^{-1}\left(b - Ax^n\right). \tag{3.9}$$

For this, a first experiment is carried out to find a simple layout for which the preconditioned Richardson iteration operator is contracting. Then, a convergence map is built.

Layout and scale robustness

The convergence of Algorithms 8 and 9 is guaranteed provided that, among other conditions, the solution algorithm is the preconditioned Richardson method and that the preconditioned iteration matrix is a contracting operator [7], that is

$$\rho(I - M^{-1}\tilde{A}) < 1. \tag{3.10}$$

The question is now to which extent this assumption is true in practice, or alternatively, whether the ORAS-preconditioned Richardson operator is always contracting in practice. Whereas the convergence of the Richardson iteration on the non-overlapping preconditioned Helmholtz problem has been analyzed in the continuous case, the convergence of the overlapping preconditioner to solve the discretized problem remains unknown [11]. Owing to the definition of the ORAS preconditioner (1.24), intuition expects that this assumption is more likely to be true as the number of subdomains is reduced to 1. An attempt is made at finding a domain decomposition configuration (*i.e.* number of subdomains, layout) for which the iteration operator is contracting. While an analysis of the eigenvalues of $I - M^{-1}\tilde{A}$ would be necessary to rigorously show the contracting nature of the operator, a more practical approach is taken here.

The layout of the subdomains is the parameter of this experiment. We consider here the checkerboard layouts described by the tuple (N, M), where N is the number of subdivisions along the width, and M is the number of subdivisions along the height.

For various layout configurations, 100 iterations of gradient descent (amounting to 200 system resolutions) with step size 5 are performed on the linearized inverse Helmholtz problem at 1 Hz. The forward and adjoint fields are calculated with the preconditioned Richardson method up to convergence, with a tolerance on the relative residual of 10^{-6}. The number of preconditioner applications to reach the 100 iterations of gradient descent is recorded, as well as whether there has been a solver failure due to divergence. Figure 3.1 shows the amount of preconditioner iterations to perform 100 gradient descent iterations; an absence of data indicates that a forward or adjoint solver failure occurred during the inversion.

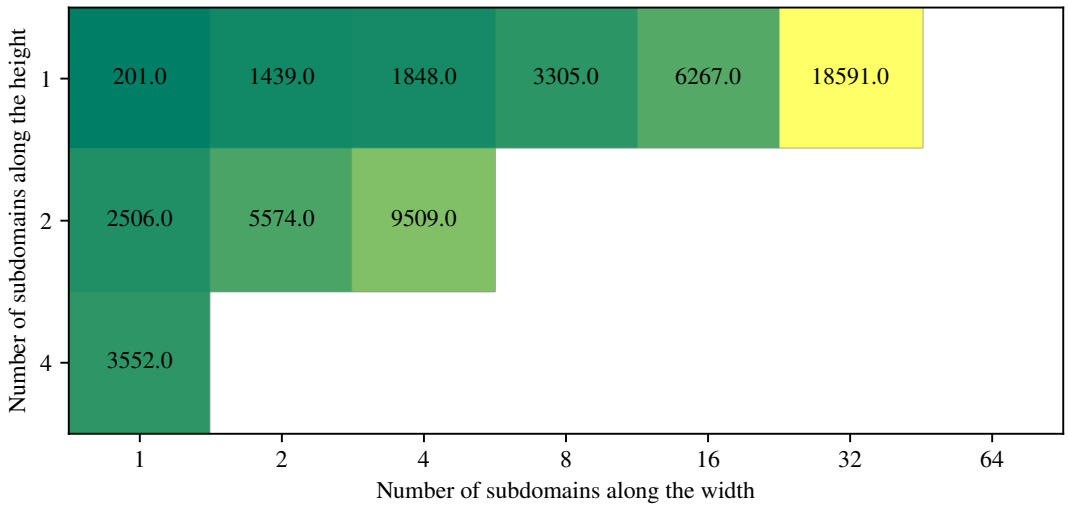


Figure 3.1: Number of preconditioner applications to perform 100 gradient descent iterations using Richardson iterations down to convergence for the resolution for the forward and adjoint fields, for the linearized inverse problem at 1 Hz in the Marmousi case. Absence of data indicates that a solver failure occurred during inversion. The convergence criterion is the relative residual reducing to 10^{-6}.

Figure 3.1 implies that while the ORAS-preconditioned unscaled Richardson method can be relatively robust to an increase in number of subdomains for given layout families, such as $(N, 1)$, its ability to produce a contracting iteration operator breaks down as the number of subdomains increases. This result motivates the transition to other iterative methods for the resolution of the forward and adjoint problems for scale-robustness. Good candidates are algorithms that apply an adaptive step size in the search direction, such as the Minimized Residual or GMRES.

Sequential multi-step one-shot convergence maps

Now that a set of partition layouts for which the ORAS-preconditioned Richardson operator is contracting has been found, the behavior of the convergence of the sequential multi-step one-shot

method (Algorithm 10) is empirically observed using a convergence map. For a given parameter space, cost metric and a convergence criterion, the convergence map associates the cost to reach the convergence criterion for each member of the parameter space.

The parameter space of the experiment is described by (k, τ), where k is the multi-step one-shot parameter, and τ is the gradient descent step size. For each member of the parameter space, 100 iterations of the fixed step gradient descent are carried out on the linearized inverse Helmholtz problem at 1 Hz. Convergence information (preconditioner applications, gradient norm and losses) are recorded; the convergence criterion is then applied after-the-fact to determine whether the parameter set is a convergent configuration and the cost to reach it. Convergence is considered reached for the smallest iteration number i such that

$$\frac{|g^n|}{\max_N |g^N|} < 0.1 \tag{3.11}$$

is true for all n such that $n \geq i$, where g^n is the gradient at iteration n. The cost metric is the number of application of the ORAS preconditioner, which corresponds to the number of iterations of the

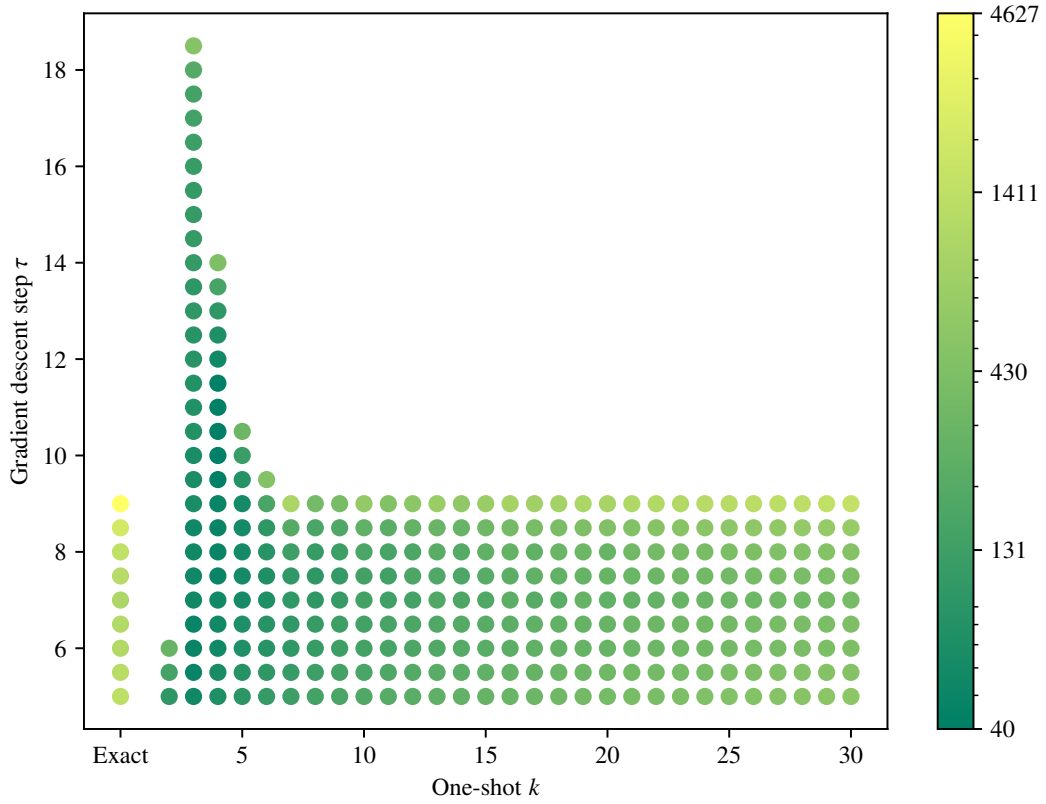


Figure 3.2: Cost to reach convergence as a function of the multi-step one-shot parameter k and the gradient descent step size τ, using Richardson iterations as the linear system solver, for the linearized inverse problem at 1 Hz in the Marmousi case. The subdomain layout is $(4, 2)$ for a total of 8 subdomains. Absence of data indicates a divergence in the inversion for the given configuration. The "exact" column indicates the cost using Algorithm 7 with a tolerance of 10^{-6} on the relative residual.

linear system resolution algorithm. This methodology is applied on the layout $(4, 2)$, for which the preconditioned unscaled Richardson iteration is contracting. Figure 3.2 shows the cost to reach convergence as a function of k and τ.

Let us denote the maximum allowable gradient descent step size $\tau_{\max}$, for which gradient descent with fixed step τ converges when

$$\tau \leq \tau_{\max}. \tag{3.12}$$

The results show that this value is a function of k. Three behavioral regions can be distinguished. For high values of k, $\tau_{\max}$ tends towards its value obtained in the column "exact"; since each iteration of the Richardson algorithm is contracting, the norm of the residual decreases, and therefore, the values of the waveform and adjoint fields approach those that would be obtained with an exact resolution. For low values of k, $\tau_{\max}$ falls drastically under the "exact" maximum step threshold; intuitively, not iterating enough on the waveform and adjoint fields leads to an incorrect gradient, and this can be mitigated by taking a smaller step so as to increase the number of iterations density in the δm space. Finally, we notice an intermediate regime in behavior of $\tau_{\max}$ between the lower k drop and the higher k plateau; a peak in the maximal gradient descent step size which is well above the higher k plateau. Conjectures about this phenomenon are discussed in a later section.

Let us denote by C the cost required to reach the convergence in the inversion. The results clearly highlight the performance merits of adopting one-shot optimization, at least for this combination of gradient descent with the preconditioned Richardson iteration. In the region of higher values of k (10 and onwards), decreasing the value of k decreases C for all steps τ. For the intermediate region (k ranging from 3 to 9), the reduction in cost with respect to the "exact" forward/adjoint resolution is not only significative, but also robust to the different values of τ.

Convergence history in each map region

In the context of Figure 3.2, three regions of gradient descent were distinguished. We seek to observe the behavior of the optimization variables (*e.g.* the losses, the gradient norm, the L^2 norm of δm) as a function of the number of preconditioner applications or gradient descent iterations for specifically chosen key configurations, in each of the three regions. Such a graph is referred to here as a convergence history. In order to better understand and get more insight on how multi-step one-shot optimization affects convergence, the history is plotted for different values of k.

The values of interest are the exact loss and the one-shot loss. At each gradient descent step, the exact loss is calculated using (1.37), the system solve is performed using an LU decomposition and does not influence the preconditioner count as well as the current value of δu. The one-shot loss is defined as

$$J(\delta u) = \frac{1}{2} \left[P_\Gamma \delta u - \delta d \right] \cdot \left[P_\Gamma \delta u - \delta d \right], \tag{3.13}$$

and is evaluated using the *current value* of δu (that is, δu^n in Algorithm 10) without performing any additional iterations of the resolution algorithm. The losses are given relative to their initial value.

Convergence histories are plotted for various values of k (2, 3, 9, 25) at a gradient descent step size of 6, which is a converging step size for all the values of k (Figure 3.2). Figure 3.3 shows the convergence history of the described configurations.

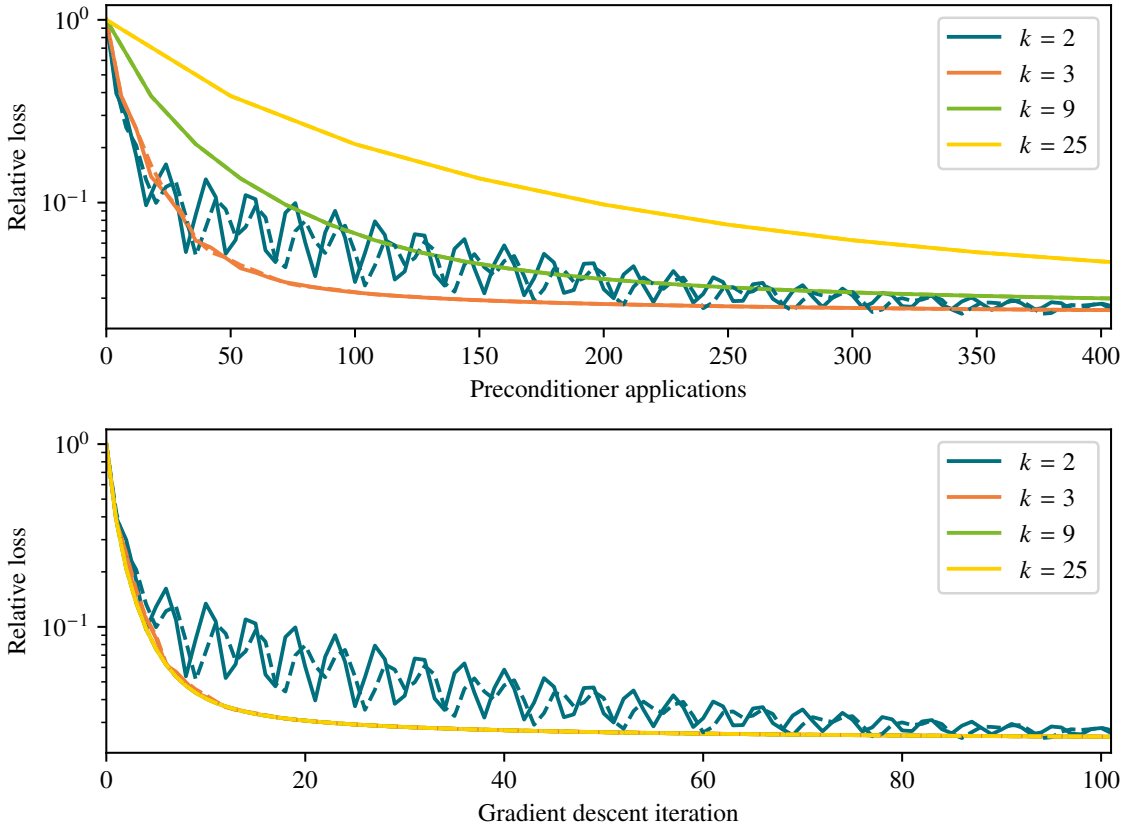


Figure 3.3: Convergence histories for the sequential multi-step one-shot method with Richardson used as the linear system solver in the one-shot scheme, for the linearized inverse problem at 1 Hz in the Marmousi case with a subdomain layout of $(4, 2)$. The optimization algorithm is gradient descent with step size 6. The dashed curves indicate the one-shot estimated loss. The upper graph is the relative loss as a function of preconditioner applications, and the lower graph is the relative loss as a function of the gradient descent iteration.

The lower graph of Figure 3.3 shows that provided that k is sufficiently high (here $k > 2$), then the overall trend of the relative loss as a function of the gradient descent iteration is mostly independent of k. The implication of this observation would be that low values of k may be chosen without degrading gradient descent convergence, thus making the convergence process cheaper in terms of preconditioner applications: this is precisely what is observed in the upper graph of Figure 3.3. The upper graph of the figure illustrates the clear potential gain of the multi-step one-shot methods: by choosing lower values of k, the relative loss can be lowered much faster for a given cost in preconditioner applications. Both views of the history show that in the case of $k = 2$, there are relatively large oscillations in the loss, which have an average value higher than the loss reached with $k = 3$. In practical situations and depending on the particular convergence criterion, these oscillations could effectively delay the reach of the criterion, making the performance poorer. These oscillations can be explained by the fact that for $k = 2$, the step size of $\tau = 6$ is near the maximal step size; their period being more than two gradient descent iterations, however, is less evidently explained. The estimated loss follows the exact loss quite well for $k \geq 3$. In the case of

$k = 2$, the estimated loss follows the trends of the exact loss but with a strong lag.

For three different values of k, we seek to observe how the evolution of the loss changes as τ approaches the maximum allowable step size $\tau_{\max}$. For each k, 100 iterations of gradient descent are performed and the exact loss and estimated loss are recorded. Figure 3.4 shows the evolution of the relative loss as a function of the gradient descent iteration, for $k = 2, 3, 25$. The step chosen as the upper limit is the highest allowable one found in Figure 3.2.

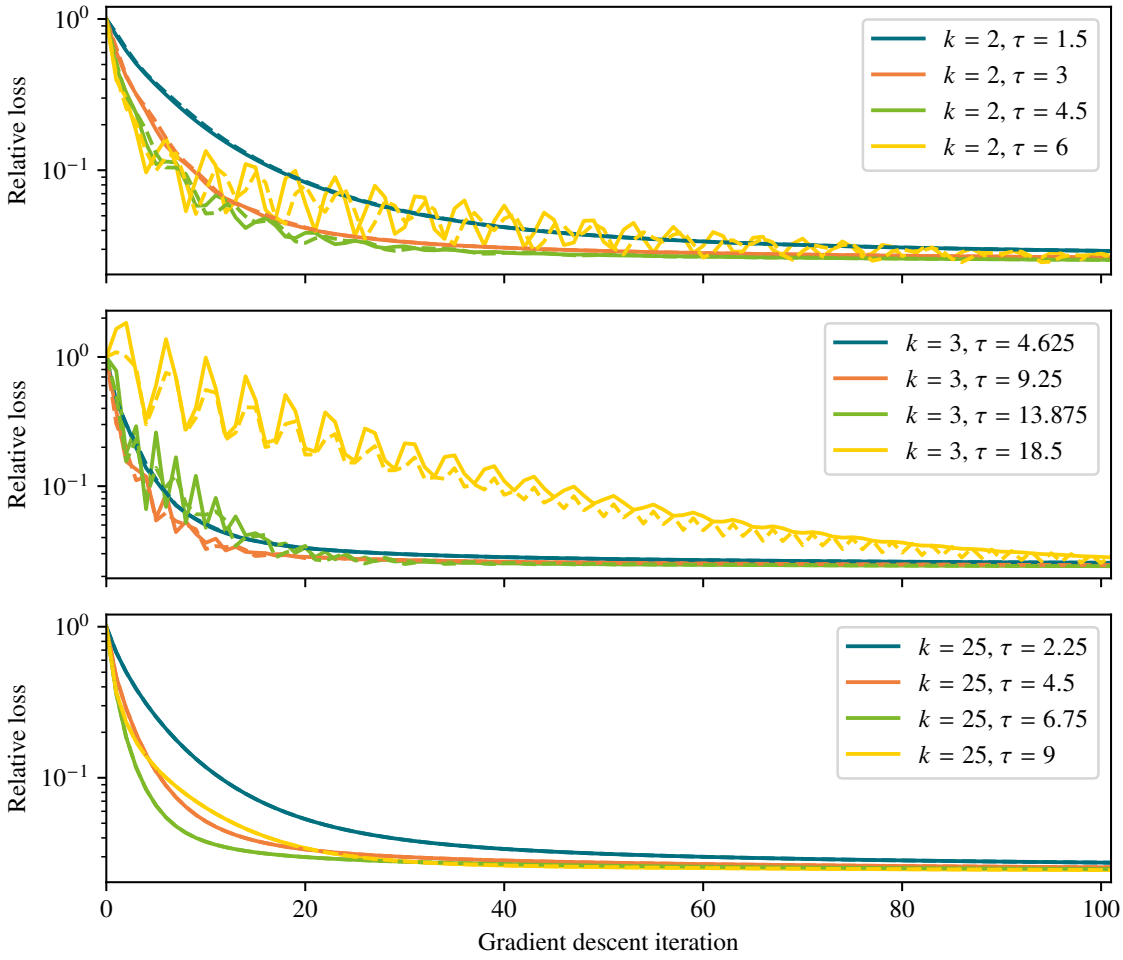


Figure 3.4: Convergence histories for the sequential multi-step one-shot method with Richardson used as the linear system solver in the one-shot scheme, for the linearized inverse problem at 1 Hz in the Marmousi case with a subdomain layout of $(4, 2)$. The optimization algorithm is gradient descent with step size τ. The dashed curves indicate the one-shot estimated loss. The graphs indicate the histories for $k = 2$, $k = 3$ and $k = 25$, respectively.

The upper and middle graphs of Figure 3.4, corresponding to values of k in the lower and intermediate regions, show that as τ approaches $\tau_{\max}$ for a given value of k, the evolution of the loss tends to be more and more oscillatory. In particular, the middle graph ($k = 3$) of the figure indicates that even though the optimization process is convergent when τ is equal to 18.5, which

is near its maximal allowable value, the relative loss decreases suboptimally. By contrast, step sizes closer to the maximal step size allowed by the higher k regions (*e.g.* 9.25 and 13.875) offer a much faster convergence. This result implies that although the intermediate region has a very good robustness to step size, it may be still desirable to choose a step size that remains limited, for example, to the order of $\tau_{\max}$ of the higher k region or less. The lower graph of the figure displays no oscillatory behavior as τ approaches $\tau_{\max}$. These results indicate that the higher k region reacts more rigidly to the transition from a converging step size to a diverging step size, whereas lower and intermediate k regions, as τ is increased, enter an oscillatory regime which starts sooner but is still converging, before finally diverging when $\tau > \tau_{\max}$.

Gradient evolution as a function of k

In order to witness how limiting the Richardson iteration count to k in multi-step one-shot affects the gradient, we seek to observe how far from the exact gradient the estimated one-shot gradient is. Denoting g as the estimated gradient and g_{ref} the exact gradient at a given step, we characterize the discrepancy between them using three metrics: the relative magnitude $||g||/||g_{\text{ref}}||$, the relative error $||g - g_{\text{ref}}||/||g_{\text{ref}}||$, and the angle between the two. The angle between the negative gradient and the step direction is an error measure of interest here for the convergence of the optimization. According to [24], a consequence of Zoutendijk's theorem is that granted sufficient regularity conditions, a necessary condition for convergence of any optimization scheme respecting Wolfe's conditions is that the exact gradient and the step direction make an angle that does not exceed 90° for all steps.

For several values of k, 100 iterations of gradient descent with a step size of 9 are performed and the measures of the difference between the exact and estimated gradient descent are performed. The exact gradient g_{ref} is, at each step, calculated by solving for the forward and adjoint fields using a direct solver using an LU decomposition. Figure 3.5 shows the relative magnitude, relative error and angle between the estimated and exact gradient descent for $k = 3$ to 6, corresponding to the peak and the higher end of the intermediate k region in Figure 3.2. Overall, the results confirm the intuition that as k is increased, the estimated gradient approaches the exact gradient in behavior. For all the values of k tested, as iterations advance towards convergence, the error measures indicate that the gradient estimate converges towards the exact gradient. The curve behavior differs between $k \leq 4$ and $k > 4$. For $k \leq 4$, the relative magnitude of the estimated gradient is sometimes largely underestimated, then overestimated in an oscillatory manner. Similar observations can be made for the angle, although they are more pronounced specifically for $k = 3$. For $k > 4$, a more stable behavior is observed. The magnitude of the estimated gradient is always smaller than the exact gradient, and converges towards that of the exact gradient in an almost non-oscillatory behavior. The magnitude being lower than the exact gradient, as well as a relatively low angle, could be an explanation as to why larger steps are allowed. For $k = 3$, it can be seen that the estimated gradient sometimes points in a direction more than 90° apart from that of the exact gradient; this implies that if k is too low, the gradient descent may take unproductive steps in the sense of Zoutendijk's condition. The fact that the estimated gradient in the case $k = 3$ is the only one to occasionally reach 90° may be an element of answer as to why this value is the lower k limit allowing convergence for the step size of 9.

The behavior of the curves for $k > 4$ in Figure 3.5 gives the impression that the estimated

gradient "lags" behind the exact gradient, and that the one-shot parameter k quantifies some sort of strength by which the estimated gradient follows the exact gradient through the descent. The validity of this mental model may not be confirmed with Figure 3.5 alone; since the one-shot estimation of the gradient is used for optimization, the series of reference gradients is not the same and several histories for different k are not comparable. We seek to perform a complementary experiment that investigates this phenomenon and the degree of validity of this mental model by using the exact gradient descent as a reference and measure the error that the one-shot estimated gradient makes with it, assuming that the previous forward and adjoint states were obtained exactly.

For several values of k, 100 iterations of the exact gradient descent (Algorithm 7) are performed. At each iteration n, a tentative next gradient g^n, which is not used for the optimization, is calculated along the exact gradient g^n_{ref} using sequential multi-step one-shot with parameter k. An important aspect of this experiment is that at all times, all previous states are considered exact, the tentative

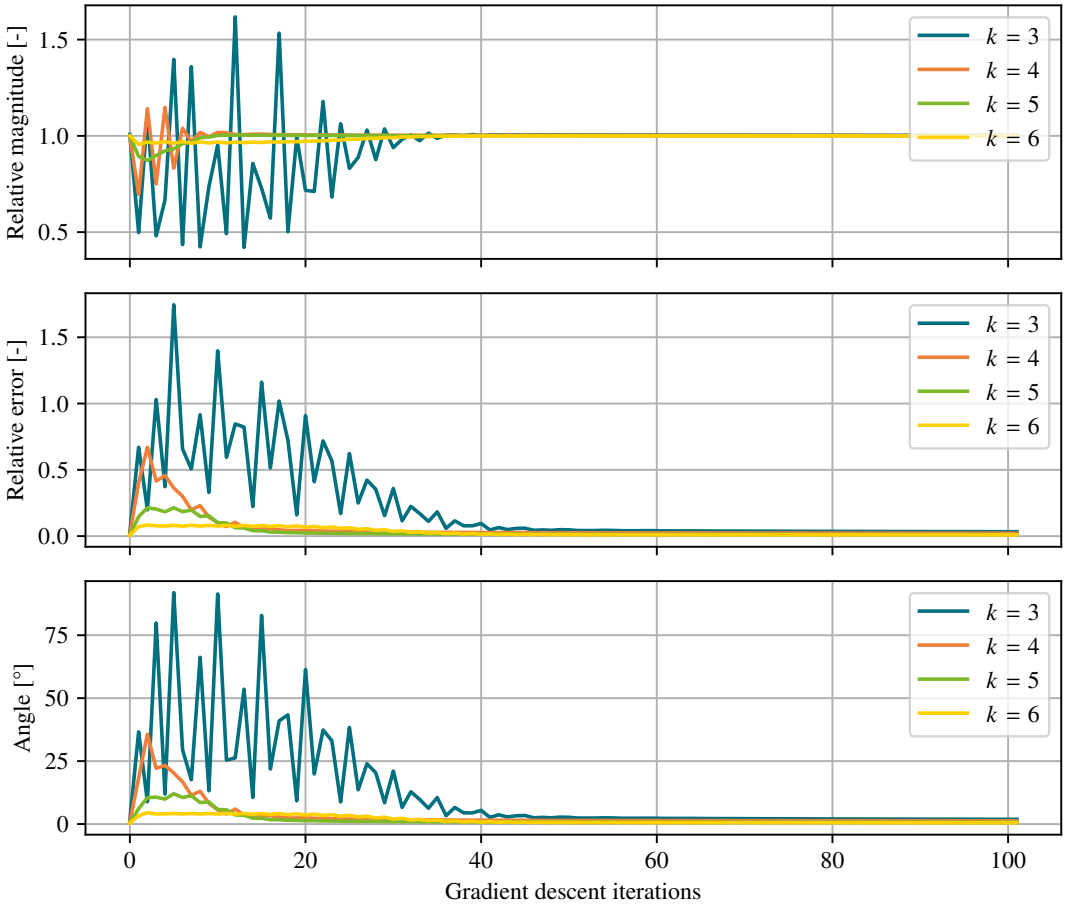


Figure 3.5: Evolution of the relative magnitude, relative error and angle of the estimated one-shot gradient with respect to the gradient calculated exactly using LU decomposition, during one-shot inversion, for the linearized inverse problem at 1 Hz in the Marmousi case with a subdomain layout of $(4, 2)$. Richardson is used as the iterative linear system solver in one-shot, and the optimization algorithm is gradient descent with step size 9. The graphs indicate the histories for $k = 3$ to 6, respectively.

gradient g^n calculated at step n can be seen as a gradient that would be obtained if one were to stop, at iteration n, solving the systems exactly and instead switching to the one-shot formalism. The relative gradient residual defined as

$$r = \frac{\left\| g^n_{\text{ref}} - g^n \right\|}{\left\| g^n_{\text{ref}} - g^{n-1}_{\text{ref}} \right\|} \tag{3.14}$$

is recorded, alongside the relative magnitude and the angle, as done in the previous experiment.

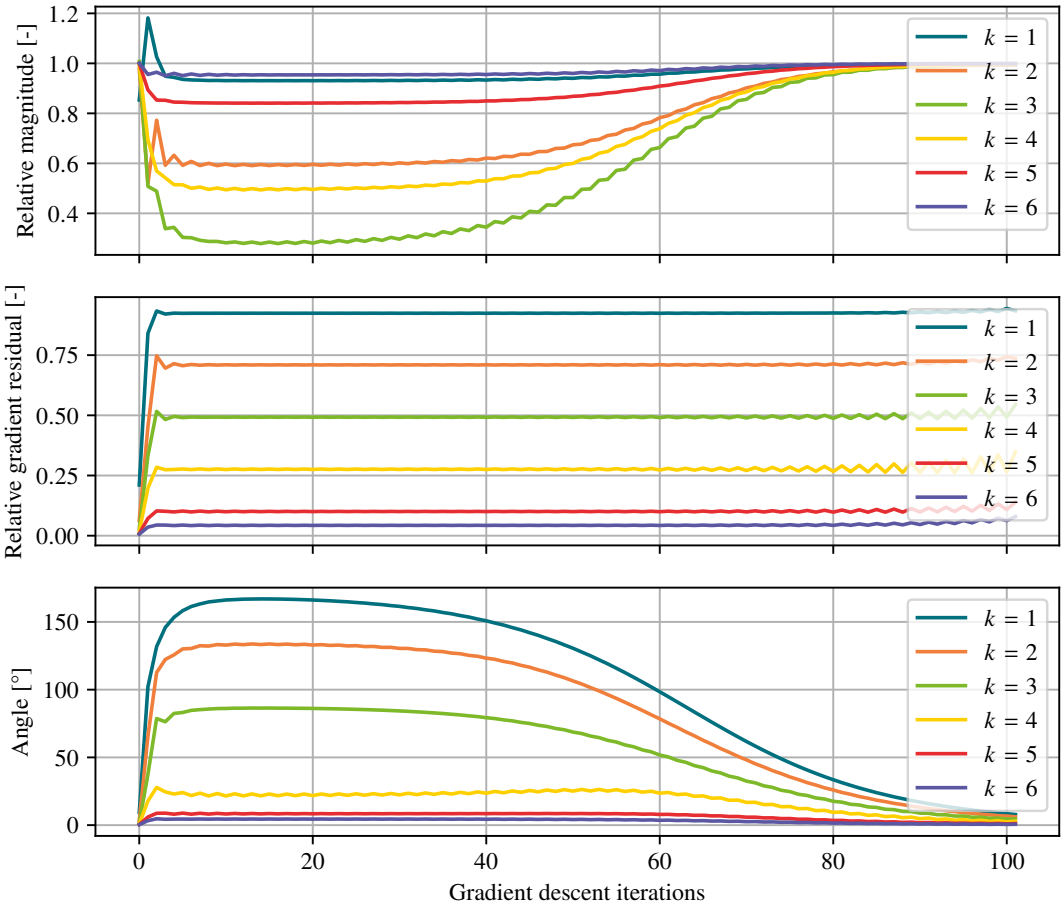


Figure 3.6: Evolution of the relative magnitude, relative gradient residual and angle of the tentative estimated one-shot gradient with respect to the gradient calculated exactly using LU decomposition, during exact inversion, for the linearized inverse problem at 1 Hz in the Marmousi case with a subdomain layout of $(4, 2)$. Richardson is used as the iterative linear system solver in one-shot, and the optimization algorithm is gradient descent with step size 9. The graphs indicate the histories for $k = 1$ to 6, respectively.

Figure 3.6 shows the relative gradient residual, relative magnitude and the angle between the exact next gradient and the tentative one-shot next gradient, as a function of the gradient descent iteration for a step size of 9. The relative residual graph shows that, for a given gradient descent iteration, the residual on the gradient decreases with a monotonous tendency as k is increased. The

growing oscillations in the curves at high iteration counts may be explained by the denominator of r tending towards zero and amplifying the relative residual when k is not large enough. The angle graph shows similar results on the angle that the two gradients make; as k is increased, the tentative one-shot gradient tends more and more towards the direction of the exact gradient. A notable result in the relative gradient residual graph for this particular step is that after a given amount of gradient descent iterations, the relative gradient residual has a constant trend lasting until the instabilities of the high iteration counts. In the angle graph, the angle of the $k = 3$ one-shot gradient is seen able to follow the exact gradient just within an angle of 90°, whereas the $k = 2$ one-shot gradient is out-of-phase by more than 90° with the exact gradient.

Figure 3.7 shows the results for the similar experiment with a gradient descent step size of 6. Both graphs display a tendency for the gradient residual and the angle between the estimated one-shot and exact gradients to decrease as k increases for almost all iterations. Interestingly, this

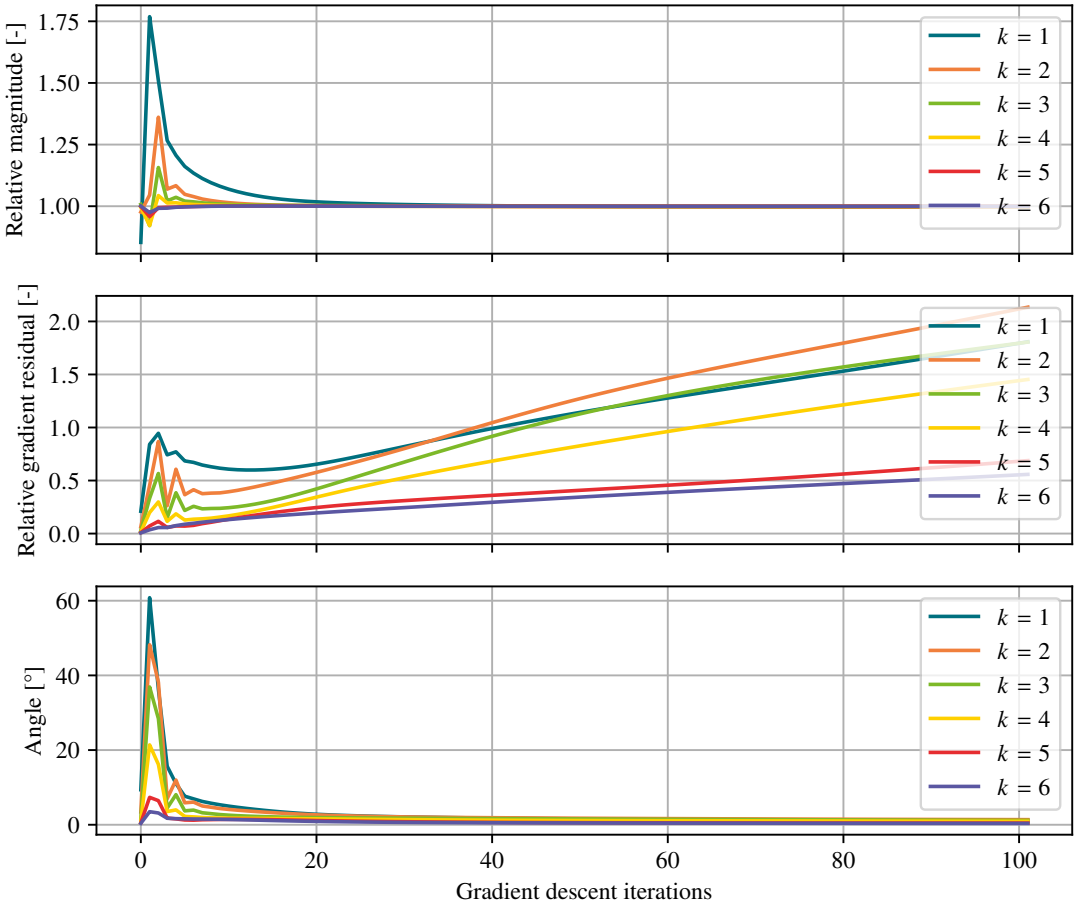


Figure 3.7: Evolution of the relative magnitude, relative gradient residual and angle of the tentative estimated one-shot gradient with respect to the gradient calculated exactly using LU decomposition, during exact inversion, for the linearized inverse problem at 1 Hz in the Marmousi case with a subdomain layout of $(4, 2)$. Richardson is used as the linear system solver, and the optimization algorithm is gradient descent with step size 6. The graphs respectively indicate the histories for $k = 1$ to 6.

is not the case for a few iterations around iteration 5, as $k = 5$ provides a slightly lower residual and angle than $k = 6$; the trend remains otherwise the same for almost all gradient descent iterations. A few differences from the case where $\tau = 9$ are noticed. For all values of k tested, the angle between the estimated and exact gradient tends to zero rapidly, and the relative residual curves increases along the iterations.

The differences in behavior of the curves between Figures 3.6 and 3.7 can be attributed to the step size chosen for the elaboration of those graphs. Figure 3.8 shows the angle between two successive exact gradients in the optimization, for different values of the step size ranging from 6 to 9. The figure shows that for $\tau = 9$, the angles between the successive gradient starts from above 100° and plateaus just below 175° for a significant portion of the experiment. A large value of the angle indicates that the gradient almost does a full direction reversal at every iteration and implies a quasi-oscillatory behavior due to τ being larger than the characteristic length of the loss well. On the contrary, the evolution of the angle for $\tau = 6$ starts at a much lower angle and reduces quickly with relatively little oscillatory behavior. In the case of $\tau = 6$, since the exact gradient does not change direction by large angles, the angle error between the exact and estimated one-shot gradients in Figure 3.7 is small; in addition, since the exact gradient does not turn back during optimization, the estimated one-shot magnitude does not decrease sufficiently fast to follow it, and the norm of the residual gets amplified by the low value of $\left\|g^n_{\text{ref}} - g^{n-1}_{\text{ref}}\right\|$, which could be causing the increase of the relative gradient residual for higher iteration counts. In the case of $\tau = 9$, the steps taken are too large with respect to the length of the well, and the exact gradient turns back strongly and often; in Figure 3.6, this translates to the plateaus in the angle and relative gradient residuals. Figure 3.8 also helps explain the different behaviors in relative magnitude between Figures 3.6 and 3.7. When the exact gradient does a direction reversal during optimization in the case of $\tau = 9$, the estimated gradient goes through intermediate states that are lower in magnitude than the exact gradient, in order to follow the exact gradient in its direction reversal, as k is increased. In the case of $\tau = 6$, and

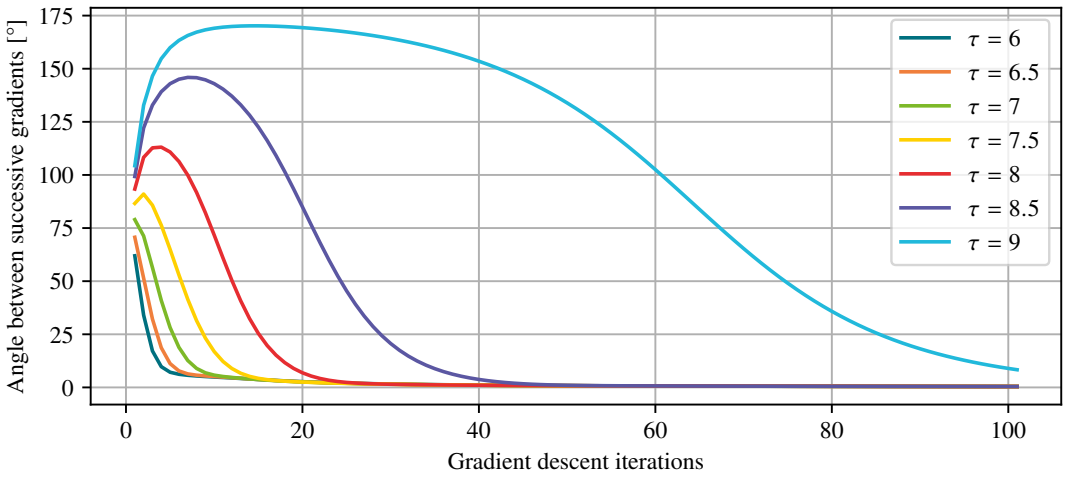


Figure 3.8: Evolution of the angle between successive gradients during inversion with gradient descent, for the linearized inverse problem at 1 Hz in the Marmousi case with a subdomain layout of $(4, 2)$. The gradients are calculated exactly using LU decomposition. Several step sizes τ are shown.

for low values of k ($k < 4$) the magnitude of the estimated gradient is higher than that of the exact gradient, as the estimated gradient has to follow the shortening exact gradient during optimization; for high values of k, the magnitude of the estimated gradient overshoots downwards.

Although no theoretical analysis was carried out, empirical evidence suggests that, at least in the tested scenarios and using Richardson as the forward/adjoint solver, it is useful to think about the multi-step one-shot gradient as following the exact gradient with a certain lag or momentum. The degree to which the estimated gradient is able to follow the changes in exact gradient during optimization is almost always increased when k is increased. Figure 3.9 illustrates, based on empirical observation of the gradient behavior of this section, the possible states that the estimated gradient could assume for different values of k in a fictional 2-dimensional one-shot inversion scenario with gradient descent.

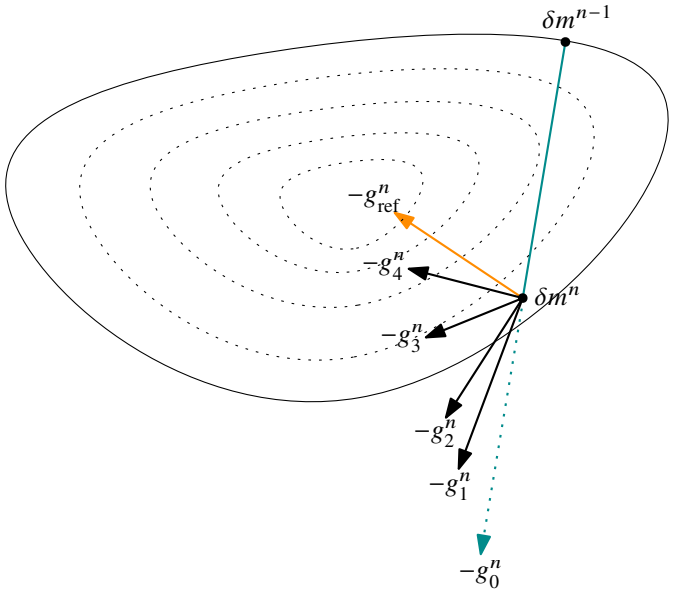
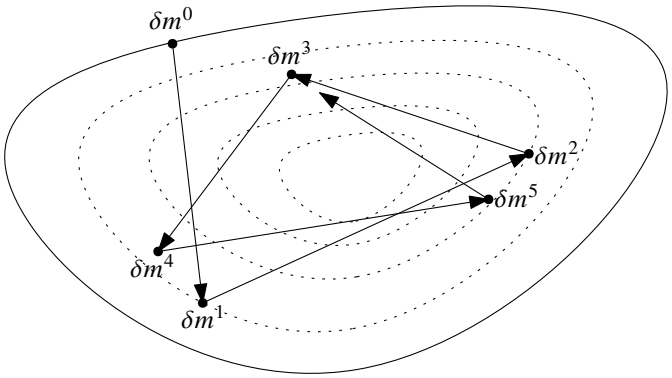


Figure 3.9: Visualization of the evolution of the k-step one-shot estimation of the gradient g^n_k with respect to the exact gradient g^n_{ref} in a fictional 2-dimensional inversion scenario with gradient descent of step size 1. δm^n is the optimization variable at step n of the optimization; the black splines indicate the isolosses. The estimation of the gradient passes through a succession of intermediate states to approach the exact gradient.

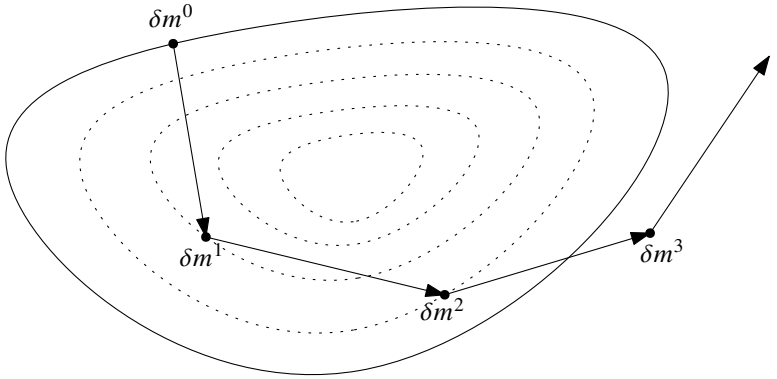
This mental model of the estimated gradient behavior allows formulating qualitative explanations for several observations made throughout this subsection. For example, the heightened step size robustness of intermediate k region, witnessed in Figure 3.2, could be attributed to the intermediate states of the estimated gradient having lower magnitudes than the exact gradient when following it during its direction reversal, all-the-while the angle between both gradients remains sufficiently low to allow for convergence. In the case of $k = 4$ for example, Figure 3.6 indeed shows that the angle remains at about 25°, while the relative magnitude can be divided by as much as 50% for some iterations.

Similarly, using this mental model, the lower k region could be attributed to the estimated gradient not being able to follow the changes in direction of the exact gradient sufficiently fast,

causing a drift of the optimization variable and thus divergence. The low frequency oscillations observed for low k curves in Figures 3.3 and 3.4 may, likewise, be attributed to the gradients being able to follow the exact gradient sufficiently fast to not diverge, while being too slow such that a revolution motion around the well occurs. Figure 3.10 illustrates in a fictional 2-dimensional scenarios two different motions that are rendered possible by multi-step one-shot inversions with a sufficiently low k, under these assumptions. Figure 3.10b shows an unstable and diverging motion around the well which could be caused by an insufficient value of k for the given step size; the estimated gradient is not able to follow the exact gradient sufficiently fast. Figure 3.10a shows a stable and convergent revolution motion around the well, which could be obtained by increasing k.



(a) Converging motion, that could cause low frequency oscillation in metrics.



(b) Diverging motion.

Figure 3.10: Visualizations of the possible motions of the optimization variable δm in a fictional 2-dimensional inversion scenario with multi-step one-shot gradient descent of step size 1, under the assumptions elaborated in this subsection. The black splines indicate the isolosses.

Further mathematical analysis which is outside the scope of this work would be necessary to confirm or infirm the validity of this mental model. In particular, this behavior is not obvious, since the gradient is obtained by not only performing the piecewise product of two fields, with the second depending on the inexact value of the first. Nevertheless, this interpretation of the results motivates the proposition of subsequent schemes and recommendations.

Conclusions

A preliminary investigation of the robustness of the preconditioned unscaled Richardson iteration has been carried out. The results of the experiments indicate that the ORAS preconditioner is not sufficient to make the Richardson iteration operator contracting as the number of subdomain increases or for more complex layout families. The transition to other iterative methods that do not make such assumptions on the preconditioned system is therefore necessary for the inversion to be robust to subdomain scaling or configuration changes.

For a working configuration, the cost to reach convergence was mapped for several (k, τ) combinations. This experiment showed that for some k, the maximal gradient descent step that can be taken can be significantly increased. In addition, for some (k, τ), the whole inversion process is cheaper by several orders of magnitude than the configuration with the "exact" solve or high values of k.

The effects of the multi-step one-shot parameter k on the gradient were investigated. In a case where a one-shot optimization is carried out, the relative magnitude and angle of the one-shot gradient with respect to the reference gradient computed with LU decomposition was found to decrease as k increases. Then, this experiment was turned inside-out: from an inversion process using the reference gradient, at each step, an estimate of the next gradient in a k-step one-shot fashion is calculated and compared with the reference. The results of this experiment showed that as k advances the gradient progressively tends in relative magnitude and angle towards that of the exact gradient. This allows thinking of the multi-step one-shot gradient during inversion as having a lag with respect to the exact gradient; the strength by which it follows the exact gradient being modulated by the parameter k.

3.2.2 Preconditioned projection methods

Section 3.2.1 showed that the ORAS preconditioner is not sufficient to make the Richardson iteration operator contracting for many complex configurations. That section concluded that the transition to iterative methods that take into account a scaling of the step is necessary to build practical schemes using the one-shot paradigm. This section now investigates the usage of preconditioned Minimized Residual method, alternatively designated as "self-scaled" Richardson [4], and preconditioned GMRES(k) as the iterative solution algorithms to substitute for `solve` in the sequential multi-step one-shot inversion (Algorithm 10).

Mirroring 3.2.1, the aptitude of those iterative methods to converge towards a working solution using an ORAS preconditioner will first be tested for several layout configurations. The cost-to-convergence maps will then be obtained for several configurations.

Layout and scale robustness

This series of experiments attempts to uncover whether the MR method and GMRES(30) are sufficiently robust to changes in domain decomposition configuration (*i.e.* number of subdomains, layout). In addition, the number of preconditioner applications between analogous configurations will be compared between the two iterative methods.

The parameter space of this experiment is the configuration of the domain decomposition and the forward and adjoint solver employed. We consider here the layouts described by the tuple (S, M) where S is the *total* number of subdomains and M is the number of subdivisions along the height. The value $M = 0$ indicates that the METIS partitioner is employed. The solver is either the ORAS-preconditioned MR method or ORAS-preconditioned GMRES(30). For each member of the parameter space, 100 iterations of gradient descent (amounting to 200 system resolutions) with step size 5 are performed on the linearized inverse Helmholtz problem at 1 Hz. The forward and adjoint fields are calculated with the given solver up to a convergence, with a tolerance on the relative residual of 10^{-6}. The number of preconditioner applications to reach the 100 iterations of gradient descent is recorded, as well as whether there has been a solver failure due to divergence. Figures 3.11 and 3.12 show the amount of preconditioner iterations to perform 100 gradient descent iterations using the MR method and GMRES(30), respectively.

Figures 3.11 and 3.12 show that unlike the preconditioned Richardson method, inversion convergence was reached for all the tested layout configurations in both preconditioned MR and GMRES. The ORAS-preconditioned Minimized Residual and GMRES methods are much more robust to scaling of subdomains. This observation is corroborated by the conclusions of [15], which highlights the robustness of the ORAS preconditioned GMRES method. The two projection methods are thus effective candidates in the role of being iterative methods that make the sequential multi-step one-shot method (Algorithm 10) converge.

Comparing the performance of GMRES and MR for a given decomposition configuration, GMRES consistently requires less ORAS preconditioner applications than MR. In addition, the transition of a number of subdivisions along the width from 8 to 16 possibly highlights scaling difficulties in the MR method, whereas GMRES displays a more robust trend as the number of subdomains is increased. Although the superiority of GMRES with respect to MR when it comes to preconditioner applications was to be foreseen due to the minimization of the residual norm in a wide basis, the unfavorable scaling properties of MR in terms preconditioner applications as a function of the number of subdomains rules it out as a practical choice.

Sequential multi-step one-shot convergence maps

The convergence behavior of the sequential multi-step one-shot method (Algorithm 10) when using the ORAS-preconditioned MR and GMRES iterative methods is empirically observed using a convergence map.

The parameter space of the experiment is described by the iterative method, being one of the ORAS-preconditioned MR or GMRES(k), the number of subdomains, as well as (k, τ), where k is the multi-step one-shot parameter and τ is the gradient descent step size. The subdomain layout is automatically generated with METIS to reduce the size of the parameter space. For each member of the parameter space, 100 iterations of the fixed step gradient descent are carried out on the linearized inverse Helmholtz problem at 1 Hz, using the sequential multi-step one-shot method (Algorithm 10). Convergence information (preconditioner applications, gradient norm and losses) are recorded; the convergence criterion is then applied after-the-fact to determine whether the parameter set is a convergent configuration and the cost to reach it. The convergence criterion and cost metric is the same as in Section 3.2.1. Figures 3.13 and 3.14 show the convergence maps

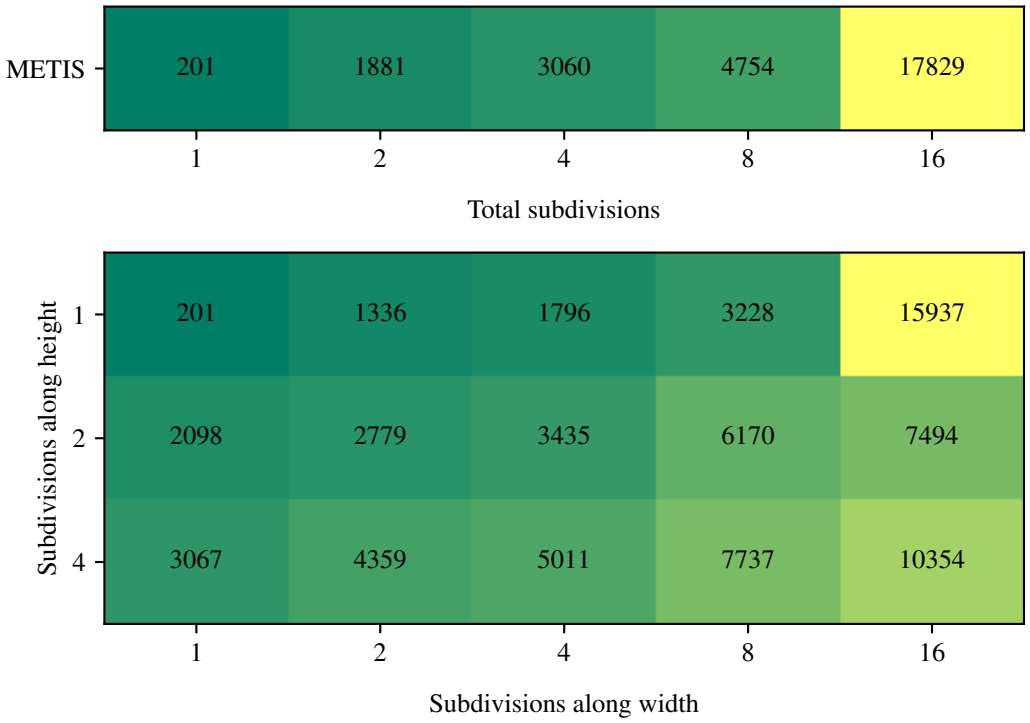


Figure 3.11: Number of preconditioner applications to perform 100 gradient descent iterations with step size $\tau = 5$ using exact resolution with Minimized Residual for the forward and adjoint fields, for the linearized inverse problem at 1 Hz in the Marmousi case.

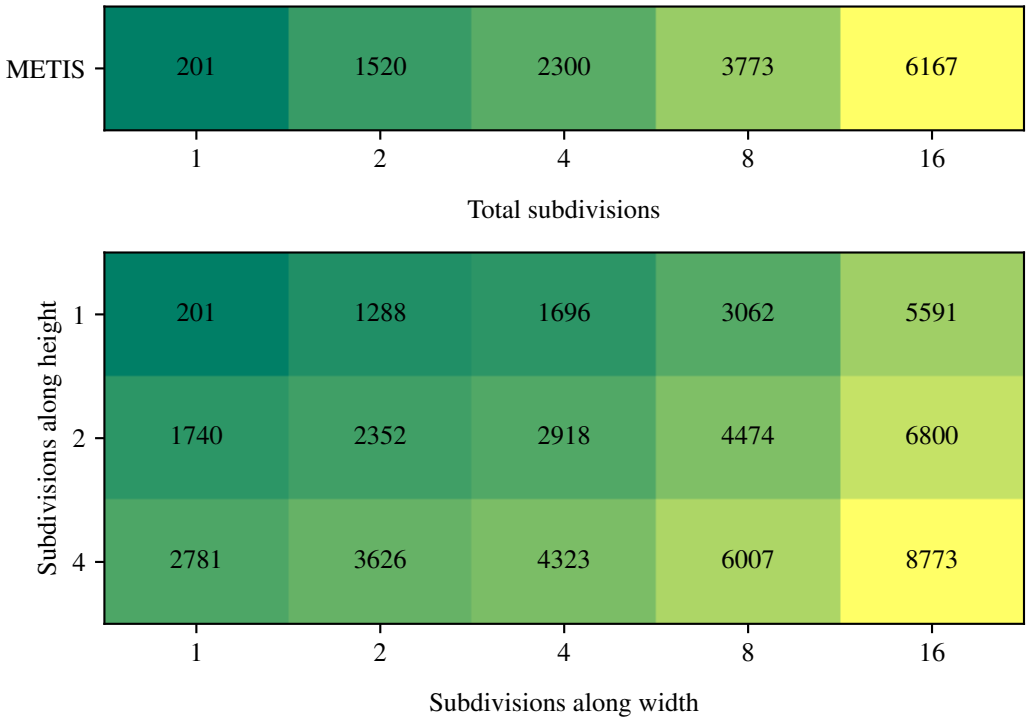


Figure 3.12: Number of preconditioner applications to perform 100 gradient descent iterations with step size $\tau = 5$ using exact resolution with GMRES(30) for the forward and adjoint fields, for the linearized inverse problem at 1 Hz in the Marmousi case.

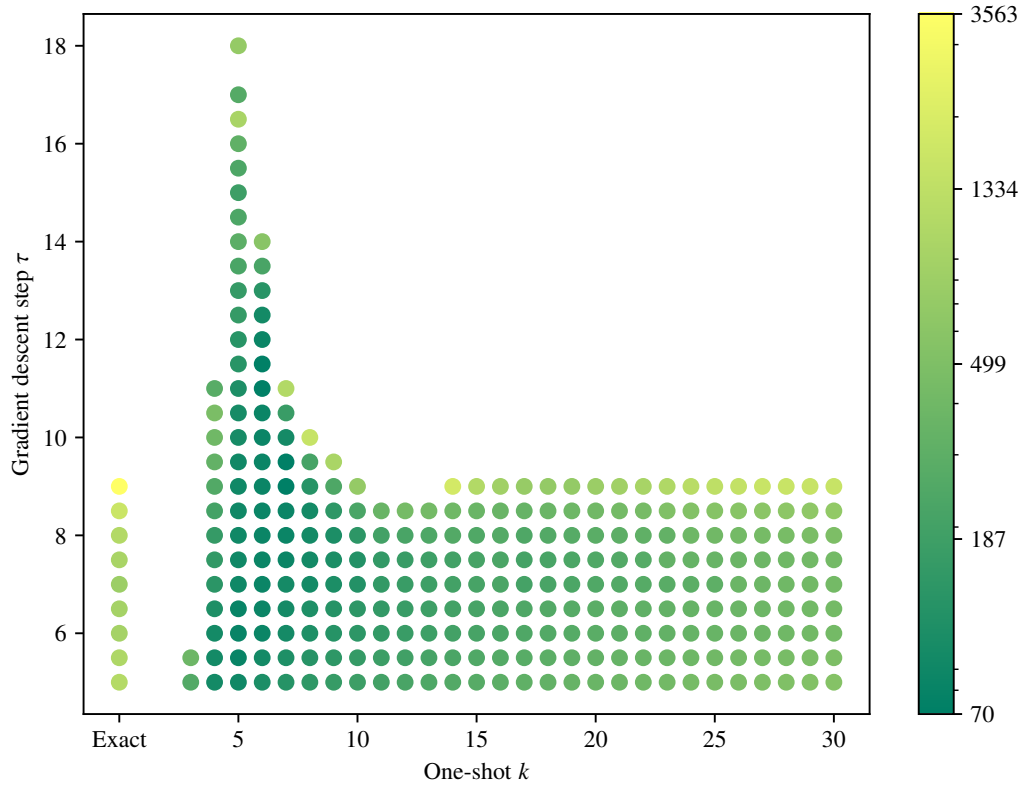


Figure 3.13: Cost to reach convergence as a function of the multi-step one-shot parameter k and the gradient descent step size τ, MR with 8 subdomains with METIS partitioning.

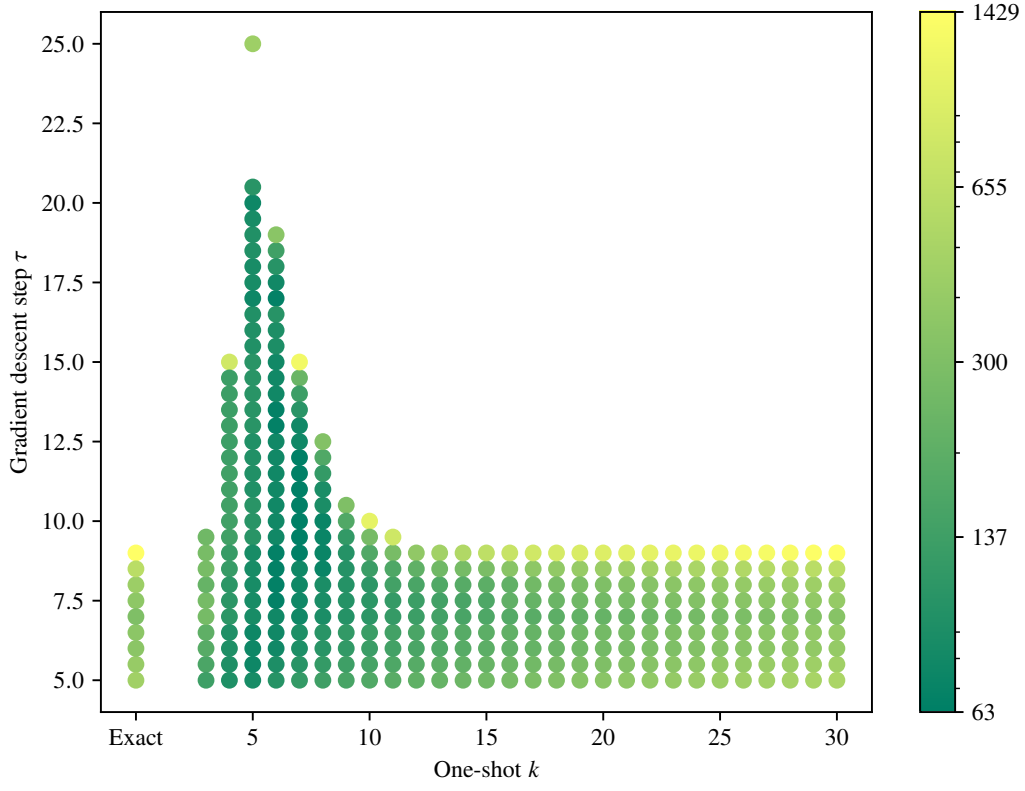


Figure 3.14: Cost to reach convergence as a function of the multi-step one-shot parameter k and the gradient descent step size τ, GMRES(k) with 8 subdomains with METIS partitioning.

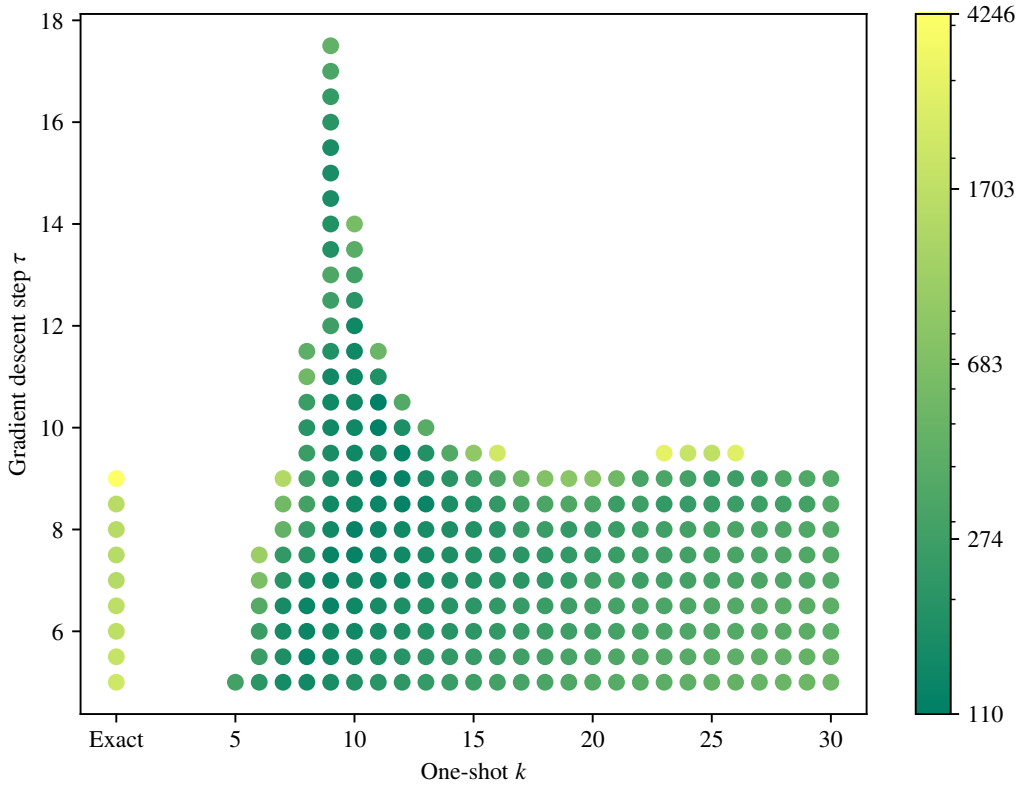


Figure 3.15: Cost to reach convergence as a function of the multi-step one-shot parameter k and the gradient descent step size τ, MR with 16 subdomains with METIS partitioning.

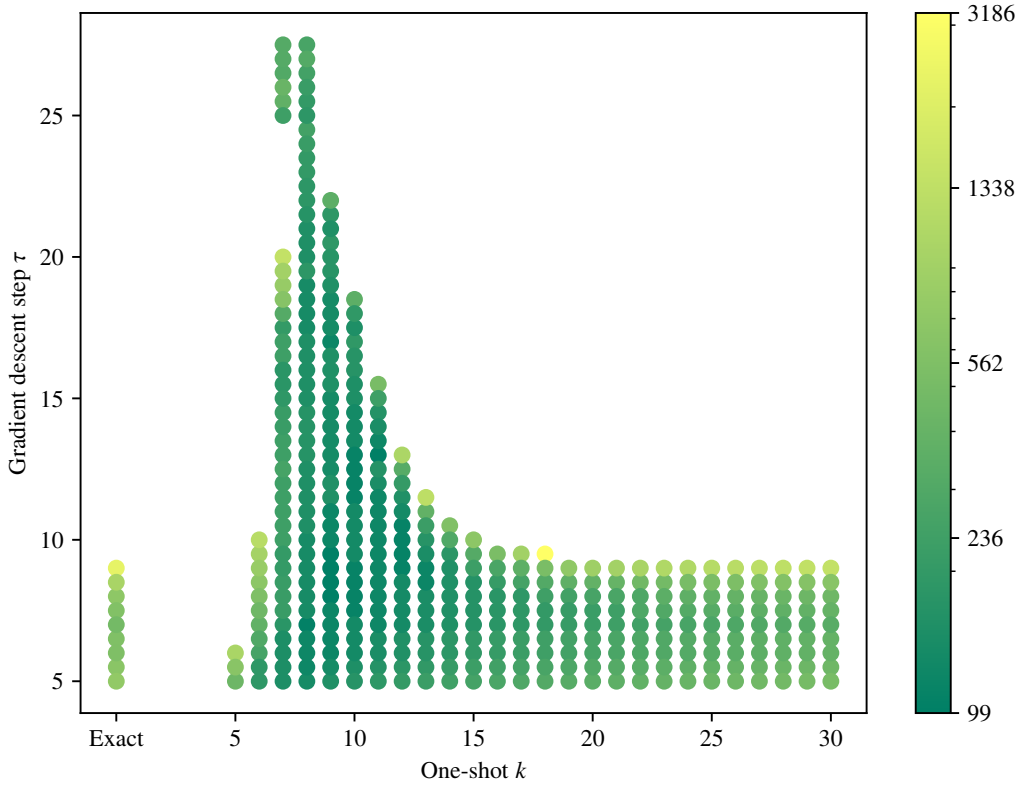


Figure 3.16: Cost to reach convergence as a function of the multi-step one-shot parameter k and the gradient descent step size τ, GMRES(k) with 16 subdomains with METIS partitioning.

of sequential multi-step one-shot method on 8 subdomains with MR and GMRES(k) respectively. Figures 3.15 and 3.16 show the same information for a domain decomposition in 16 subdomains.

Figures 3.13 to 3.16 highlight a similar trend as in 3.2. The k space can be split in three regions. The higher k region ($k > 15$ for MR, $k > 12$ for GMRES) where both the maximal step size and the cost of inversion convergence for a given step size approach the behavior of Algorithm 7; in this region, the gradient is a good substitute for the exact gradient in terms of its behavior. Both MR and GMRES(k) convergence maps display a lower k region ($k < 5$ for MR, $k > 4$ for GMRES) where the maximal gradient descent step size to achieve inversion convergence drops drastically alongside a slight increase in cost for a given step size. Finally, both convergence maps display an intermediate region ($5 \leq k \leq 15$ for MR, $4 \leq k \leq 12$ for GMRES) with a steep increase in allowed maximal step size; in both MR and GMRES, the intermediate region contains the optimal (k, τ) configuration in terms of preconditioner applications, and also shows a much higher robustness of the cost to the step size. The estimated gradient corresponding to this intermediate k region is clearly inexact, but remains effective at minimization, and allows higher step sizes through a dampening effect.

Comparing the results with MR to those with GMRES for 8 subdomains (Figures 3.13 and 3.15), the convergence map using MR displays a non-monotonous maximum step "hole" in the intermediate region (between $11 \leq k \leq 13$), whereas GMRES monotonously tends towards the asymptotic maximum step. The value of k corresponding to the peak of maximum step size is approximately the same for both MR and GMRES, however the latter convergence map displays a peak which is more robust to changes in k. In addition, the maximum step size tends to be much higher for GMRES than MR, which could indicate that the one-shot estimated gradients with GMRES pass through intermediate states with lower amplitude than with Richardson and MR.

The transition from 8 subdomains to 16 subdomains has several effects on both methods. The peak location is offset to greater values of k: as the number of subdomains increases, the quality of the ORAS preconditioner decreases, and more iterations are needed to transmit information over a given distance in the domain. An increase in number of subdomains also induces a widening of all three regions of the convergence maps, which increases the robustness of the multi-step one-shot scheme to suboptimal choices of k.

Conclusions

The preliminary analysis showed that the ORAS-preconditioned MR and GMRES can correctly calculate the waveform and adjoint fields in the context of FWI for almost any arbitrary layout configuration and number of subdomains provided that a sufficient number of iterations are performed. The iteration count needed to solve for the forward and adjoint fields scales significantly better with the number of subdomains for GMRES than MR. Whereas both MR and GMRES are contenders for the role of a "working Richardson" in the context of ORAS-preconditioned iterative FWI, the use of GMRES is clearly desirable.

Adding in the multi-step one-shot inversion and its parameter k, both MR and GMRES showed similar trends as those observed with the Richardson method. A comparison of MR with GMRES showed that the intermediate k region of GMRES tends to procure more step-size-robustness to gradient descent. In addition, the behavior is observed to approach the exact behavior monotonously

in GMRES, whereas regions of unpredictable decreases in $\tau_{\max}$ and increase of cost can be observed for MR.

Due to the favorable scaling and robustness properties of GMRES observed in this section, and the relative limited complexity overhead of the method [32], especially when the restart parameter is set to multi-step one-shot parameter k, GMRES is considered the method of choice for the practical implementation of the multi-step one-shot methods in the rest of this work.

3.2.3 Recommendations and conclusions

This section has investigated the multi-step one-shot method applied on one linearization of the problem in the frame of the fixed-step gradient descent as the optimization algorithm. Three system resolution strategies were tried: Richardson, MR, and GMRES(k).

A preliminary analysis on the ORAS-preconditioned unscaled Richardson was attempted. This algorithm was found not able to provide a contracting iteration operator for many complex layout configurations, including an increase in subdomain count. The inability of Richardson to provide a contracting iteration prompted the switch to MR and GMRES. MR and GMRES demonstrated greater robustness to the number of subdomains and to the layout configuration. GMRES was found to have superior and more desirable scaling properties than MR.

The multi-step one-shot algorithm was then tested for all three algorithms, on fixed step gradient descent. In every case three regions could be distinguished: for low values of k, an undesirable valley where the maximum allowable step $\tau_{\max}$ falls dramatically to make up for the insufficient number of iterations. For high values of k, the waveform and adjoint fields approach their exact value sufficiently for the behavior of the gradient, and thus the inversion, to become similar to the exact one. For intermediate values of k, gradient descent allows greater step sizes and reaches convergence very quickly. Among the three iterative methods, GMRES once again showed desirable properties especially in the intermediate region.

This section, although it is based on limited field experience, allows devising informal operating guidelines for the rest of this work and the beginning of a path towards a more practical one-shot inversion workflow. For the linear system resolutions, use GMRES, with a restart parameter set to k or less, for the calculation of the forward and adjoint problems. If the optimization algorithm requires gradient information which is relatively reliable, choosing a k located in the higher k region is probably necessary; if the optimization algorithm is more robust to less exact gradients, choose a k in the higher end of the intermediate region. Of course, in practice, the choice of k is not obvious *a priori* and a scheme to adaptively choose it is desirable. Similarly an adaptive scheme to control the step size would clearly be desirable.

3.3 Preconditioned one-shot

Section 3.2 has considered the behavior of multi-step one-shot methods in the context of gradient descent with fixed step size, for solving the linearized inverse problem. One of the issues of this approach is that the step size is a dimensional quantity and that its ideal scale is not only

problem-dependent; it also varies during optimization. Indeed, consider

$$\delta m^{n+1} = \delta m^n - \tau g^n. \tag{3.15}$$

Since g^n is the gradient of δm^n, the dimensionality of g^n is different from that of δm^n, which implies that τ must be dimensional and thus problem dependent. In addition, figures such as Figures 3.8 and 3.6 clearly imply that the direction and scale of the gradient changes during the inversion, and that through preconditioning, a more efficient inversion could be obtained by adapting the step direction and its scale.

In practical situations, it is desirable to employ optimization methods that scale the step direction, and possibly even choose a step direction which is not exactly the negative gradient, in order to achieve a more efficient optimization scheme. However, given that in one-shot optimization, the gradient is merely inexactly estimated, one may reasonably expect that some assumptions made by such methods are not met. The purpose of this section is to investigate such methods in the context of multi-step one-shot optimization for solving the linearized inverse problem. This section will first delve into the more traditional iterative methods for solving linear systems, which include scaling of step direction, such as the steepest descent, the minimized residual, and the conjugate gradients, when applied to the Newton equation, and explain why their inclusion into a greater one-shot optimization algorithm is not an obvious task due to the need to evaluate the gradient at a location which is not an iterate of the optimization variable. This observation will motivate the transition to techniques that only require previous iterates to build a preconditioner for the gradient.

3.3.1 Projection methods for the Newton equation

In the context of FWI using a succession of linearized inverse problems, or using the Gauss-Newton method, each step of the optimization reduces into the minimization of a convex quadratic misfit [24]. If not by fixed step descent, this minimization can be carried out by solving the Newton equation

$$\nabla_m^2 \tilde{L}(0)\delta m = -\nabla_m \tilde{L}(0) \tag{3.16}$$

for δm. The direction δm is designated as the search direction for the minimization of the non-linearized inverse Helmholtz problem. As explained in Chapter 1, obtaining the full form of $\nabla_m^2 \tilde{L}(0)$ is impractical, and therefore one has to turn to iterative methods for the resolution of (3.16) by knowing the result of applying $\nabla_m^2 \tilde{L}(0)$ to δm using finite differences. For the sake of disambiguation, let $g(\delta m)$ be the gradient of the linearized inverse problem loss (see Section 1.5.7), at δm, defined as

$$g(\delta m) = \mathrm{Re}\left\{\left[\omega^2 P_\Gamma \tilde{A}^{-1}\tilde{U}\right]^* \left[\omega^2 P_\Gamma \tilde{A}^{-1}\tilde{U}\delta m + \delta d\right]\right\}, \tag{3.17}$$

where $g(0) = \nabla_m \tilde{L}(0)$. Of course, in a one-shot optimization context, the expressions $\tilde{A}^{-1}\tilde{U}\delta m$ are practically evaluated using an appropriate iterative solver truncated to k iterations. Using the gradient of the linearized inverse problem loss, the residual r of (3.16) can be rewritten exactly using finite differences as

$$r = -g(0) - (g(\delta m) - g(0)) = -g(\delta m). \tag{3.18}$$

A natural way to introduce adaptativity in the step size is to consider the resolution of (3.16) using a linear system solver based on projection methods fit for symmetric positive definite matrices such

as the steepest descent, the minimized residual, the residual norm steepest descent, or the conjugate gradients [31]. At each iteration, for a given choice of unscaled step direction p, these algorithms update the iterate δm using

$$\delta m \leftarrow \delta m + \alpha p, \tag{3.19}$$

$$r \leftarrow r - \alpha \nabla_m^2 L(m_k) p \tag{3.20}$$

where α is a step size. In these methods, an ideal step size is obtained such that it is the one that orthogonalizes the new residual with respect to a subspace, for a given choice of step direction. Table 3.1 shows the step direction, subspace basis, and resulting step sizes for the most common projection methods for linear systems.

Method	**Direction**	**Basis**	**Step size α**
Steepest descent	r	r	$\frac{\langle r,r\rangle}{\langle Hr,r\rangle}$
Minimized residual	r	Hr	$\frac{\langle Hr,r\rangle}{\langle Hr,Hr\rangle}$
Residual norm steepest descent	H^*r	HH^*r	$\frac{\langle H^*r,H^*r\rangle}{\langle HH^*r,HH^*r\rangle}$
Conjugate gradients	p	p	$\frac{\langle p,r\rangle}{\langle Hp,p\rangle}$

Table 3.1: Summary of the step direction, residual orthogonalization basis and the resulting step size for four projection methods. r stands for the residual – in the case of the Newton equation, the negative gradient. H stands for the Hessian. In the case of the conjugate gradients, p is the direction obtained by H-orthogonalizing the new residual with the previous one.

The calculation of the step sizes of the projection methods all require the evaluation of Hr or Hp. Using finite differences, these expressions amount to obtaining $g(p)$ or $g(g(\delta m))$, which correspond to evaluations of the gradient at non-iterates. In a general context, evaluating the gradient at a non-iterate would not be considered an issue and would be considered an unavoidable cost of adding in adaptativity to the step size. In the frame of one-shot optimization where the last iterate of δu and λ are reused as initial guesses of the iterative system solution algorithms (Algorithms 8 to 10), calculating $g(p)$ or $g(g(\delta m))$ would require adding a new set of inner loops, after the evaluation of the gradient, to recompute a new gradient in an unscaled (possibly large) direction. In addition to not lending itself to a one-shot framework, the performance and convergence implications of such a scheme are unclear; since the knowledge of $g(\delta m)$ is already inexact itself, one may expect that the knowledge of $g(p)$ or $g(g(\delta m))$, themselves calculated using the one-shot paradigm, would be even more so.

Besides the natural unfitness of the projection methods in the context of one-shot optimization, Fletcher noted [12] that the performance of the conjugate gradient iteration may seriously be degraded when the step direction calculation departs from a quadratic model. Even though the misfit functional of the linearized inverse problem is quadratic, the one-shot estimation of the gradient is inexact and is also not idempotent: *i.e.* since the estimation of the gradient depends on the running value of δu and λ, evaluating the one-shot gradient at the same location may result in different estimations. Notwithstanding the above-mentioned impracticalities in the step size calculation common to all projection methods, it is likely that the linear conjugate gradient method would lead to subpar performance due to inexact gradients.

These observations on the unfitness of projection methods motivate the search of gradient preconditioning methods that explicitly do not require evaluating the gradient at non-iterates. Among such methods are quasi-Newton methods, that only require the gradient at past and current iterates to precondition the gradient.

3.3.2 Barzilai-Borwein

A class of gradient preconditioning strategies for optimization are the quasi-Newton methods. As explained in Chapter 1, quasi-Newton methods precondition the gradient by constructing an approximation of the inverse Hessian from the gradients at previous iterates. By doing so, the step direction at step p^n is given by

$$p^n = -B_n^{-1} g^n, \tag{3.21}$$

where B_n^{-1} is the inverse Hessian approximation at step n. Generally, p^n does not need to be collinear to $-g^n$, although of course, Zoutendijk's theorem [24] requires that both directions make an angle of less than 90° for global convergence provided that the Wolfe's conditions are met. Since the exact Hessian operator takes into account the local curvature of the misfit, and is used to create a quadratic local model of the misfit, one would expect that any good approximation of the inverse Hessian *appropriately* scales the gradient.

We consider here arguably one of the simplest quasi-Newton methods, consisting of a scalar approximation of the inverse Hessian, called the Barzilai-Borwein method [5].

Long and short step sizes

The Barzilai-Borwein method [5] introduces two step sizes based on the previous step $\Delta\delta m$ and the difference between the gradient at the current and previous step Δg. The so-called *short* Barzilai-Borwein step is given by

$$p^n_{\text{short}} = -\frac{\langle \Delta g, \Delta\delta m \rangle}{\langle \Delta g, \Delta g \rangle} g^n, \tag{3.22}$$

whereas the so-called *long* Barzilai-Borwein step is given by

$$p^n_{\text{long}} = -\frac{\langle \Delta\delta m, \Delta\delta m \rangle}{\langle \Delta g, \Delta\delta m \rangle} g^n. \tag{3.23}$$

Interestingly, it can be noted that the short step size Barzilai-Borwein corresponds to the scaling of the initial matrix in the L-BFGS method (Algorithm 6); however, the L-BFGS method as it is traditionally defined cannot be qualified as a generalization of the short-step Barzilai-Borwein, since the parameter m of L-BFGS must be greater or equal to 1, which implies a change in step direction.

Although the Barzilai-Borwein method does not guarantee monotonous convergence [5], Raydan [27] demonstrated the global convergence of the method for the optimization strictly convex quadratic functions. In the case of non-quadratic functions, Raydan [28] combined the method with the use of a non-monotone globalization strategy such as the Grippo-Lampariello-Lucidi condition [16] to ensure global convergence even when the function does not decrease monotonously.

Fletcher [12] investigated the Barzilai-Borwein method and compared it to the minimized residual and conjugate gradients projection methods. Although the projection methods generally yielded better performance for the minimization of well-conditioned quadratic functions, the performance of Barzilai-Borwein is much more robust to ill-conditioning than projection methods, sometimes reaching comparable performances. In a function containing small non-quadratic terms, Fletcher noticed that the unmodified Barzilai-Borwein could outperform the Polak-Ribière nonlinear conjugate gradient.

In FWI on the linearized inverse problem, the misfit functional is purely quadratic. In addition, in the context of one-shot optimization, the gradient information is inexact. The Barzilai-Borwein method and its variants appear as interesting and simple contenders to bring step size adaptativity to gradient descent in a one-shot environment.

Positive step sizes

For the optimization of convex objective functions using exact gradients, the step sizes (3.22) and (3.23) take positive values. When the function is non-convex or when the gradient information is not exact, those step sizes choices may become negative. The misfit function of the linearized inverse Helmholtz problem is convex by construction; however when performing one-shot optimization, the estimation of the gradient is inexact, and this may lead to a negative Barzilai-Borwein step size. Indeed, Section 3.2.1 highlighted that a possible behavior for the one-shot estimated gradient is that it follows the exact gradient with a certain lag in relative residual and angle; it is thus easy to imagine a scenario in which the estimated gradient grows in magnitude during descent, counter to the convexity of the well, causing the (3.22) and (3.23) to be negative. Such a scenario may for example arise after a reversal in the direction of the exact gradient. Figure 3.17 illustrates in a simple 1-dimensional case a scenario in which the one-shot estimated gradient may increase in magnitude during descent on a purely convex function.

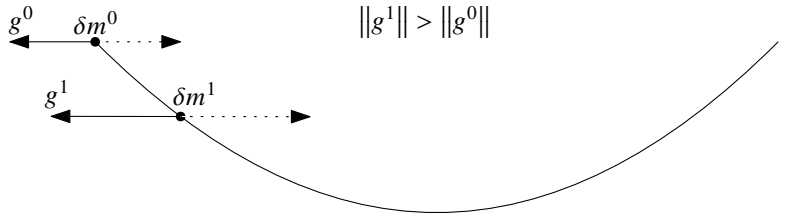


Figure 3.17: Visualization of a scenario in which the one-shot estimated gradient increases in magnitude during descent on a purely convex function. g^n and δm^n indicate the one-shot estimated gradient and the optimization variable at step n, respectively. The black curve is the function to be minimized for δm.

To remedy this, at least in the context of non-convex optimization, Dai et al. [9] investigated alternative step sizes that are positive by construction. In particular, the following step size

$$p^n_{\text{mean}} = -\frac{||\Delta\delta m||}{||\Delta g||} g^n, \tag{3.24}$$

corresponding to the geometric mean of the short and long Barzilai-Borwein step sizes, is necessarily positive. Saab et al. [30] proposed another related positive step size, in the context of optimization with momentum, which takes into account estimations of the largest and smallest eigenvalues of the Hessian matrix.

When using (3.22) and (3.23) for one-shot optimization of the linearized inverse problem misfit, a negative step size is the symptom of a gradient not being able to follow the curvature sufficiently fast, as opposed to the function being non-convex. Therefore, one may reasonably be critical as to whether forcing the positiveness of the step-size is a good strategy, since any kind of step estimation made from the estimated gradient would then be made from a picture of the landscape of the function to be optimized known to be incomplete or incorrect. Instead, a more desirable strategy would be to address the root cause of the problem by increasing the multi-step one-shot parameter k to increase the accuracy of the estimated gradient. Nevertheless, the performance of the step size (3.24) is still to be investigated for one-shot optimization.

Convergence histories for exact gradients

Due to the prevalence of projection methods to solve the Gauss-Newton equation and of more complex quasi-Newton methods to optimize the non-linearized inverse problem directly, only few examples exist of attempts to use the Barzilai-Borwein method in FWI, whether on the linearized or non-linearized inverse problem. Fu et al. [13] investigated the use of the Barzilai-Borwein method to implement a variable projection method in FWI, with some success. We seek here to gain our own first-hand experience with the Barzilai-Borwein method for the inversion of the linearized inverse problem using exact gradients.

For each step size choice among (3.22), (3.23) and (3.24), 100 iterations of the Barzilai-Borwein gradient descent are carried out on the linearized inverse Helmholtz problem at 1 Hz. The system resolution algorithm is GMRES(30). The initial model has been smoothened from the Marmousi model as described in Section 2.2.5. The waveform and adjoint fields are calculated exactly with a tolerance on the relative residuals of 10^{-6}. The initial step size is naturally a parameter of the scheme, and is set to 1. Figure 3.18 shows the evolution of the relative loss and step size as a function of the gradient descent iterations, for all three variants of the Barzilai-Borwein method. The first detail to notice on the figure is that the relative loss convergence induced by (3.23) is much less stable than the one induced by (3.22) or (3.24). Indeed, with the long step size, the relative loss occasionally jumps back up by orders of magnitude, sometimes even reaching above its initial value; on the contrary, only minor increases are noticed for the short and mean step sizes. When it comes to the evolution of the step size, all three choices display a highly oscillatory behavior. Interestingly, while the short and long step sizes display tendencies for high jumps, the peak step sizes induced by inversion with (3.24) tend to be much milder.

These experiments empirically show that the short and mean choices of Barzilai-Borwein step sizes are adequate for the solution of the linearized inverse Helmholtz problem with the parameters considered. As expected, neither (3.22) nor (3.23) yielded a step size which is negative. The long step size induced several very high jumps in the relative loss. Although those high jumps were quickly corrected in a few iterations, this correction probably requires a significant reversal of the gradient. As observed in Section 3.2.1, the one-shot estimated gradient is characterized by a lag

in its following of the exact gradient during its changes in direction, therefore it is reasonable to expect that the long Barzilai-Borwein step choice will likely be less adequate for optimization with one-shot, at least for lower values of k.

Convergence histories in one-shot

We now seek to observe how the Barzilai-Borwein step sizes fare for the inversion of the linearized inverse problem using the multi-step one-shot paradigm.

For each step size choice among (3.22), (3.23) and (3.24), and for several values of k, 100 iterations of the Barzilai-Borwein gradient descent are carried out on the linearized inverse Helmholtz problem at 1 Hz using the sequential multi-step one-shot algorithm (Algorithm 10). The initial step size is 1. The loss and the step size are recorded at each iteration. Figures 3.19a, 3.19b and 3.19c respectively show the relative loss, with respect to the initial value, and the step size, for the short, long and mean Barzilai-Borwein step sizes respectively. The results show that for all variants, the scheme fails to converge if k is too low. The short step variant converges for $k \geq 5$, the mean step for $k \geq 4$ and the long step for $k \geq 6$.

Figure 3.19b shows that the long step Barzilai-Borwein variant requires a higher value of k to produce convergent results, as expected. For values of k that ensure convergence, spikes reminiscent

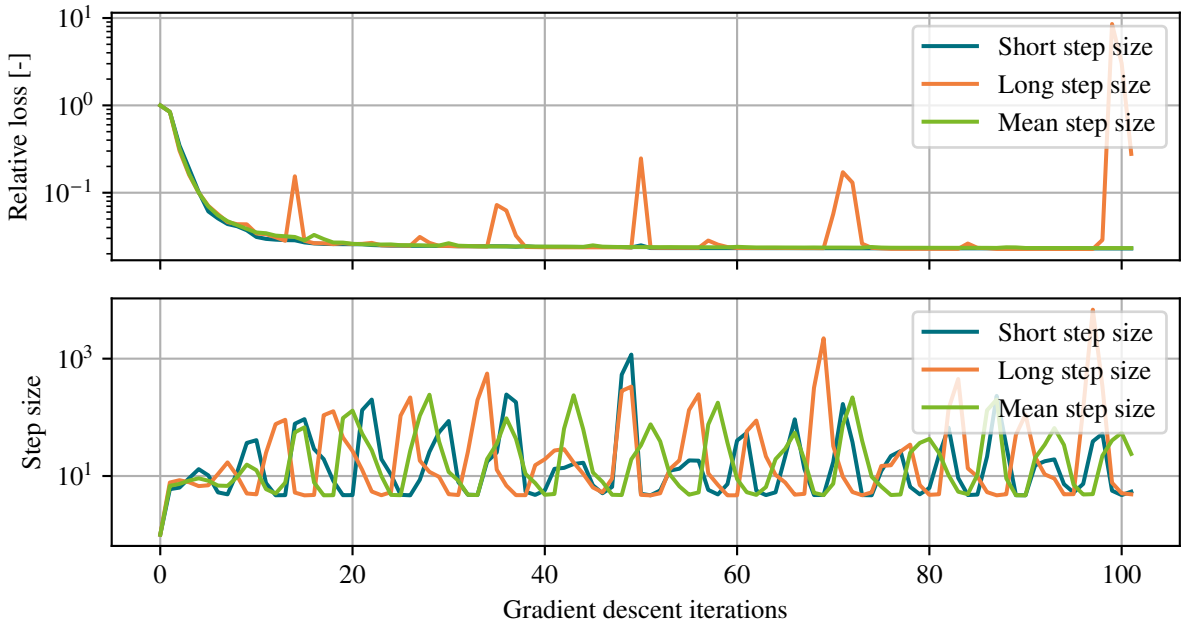
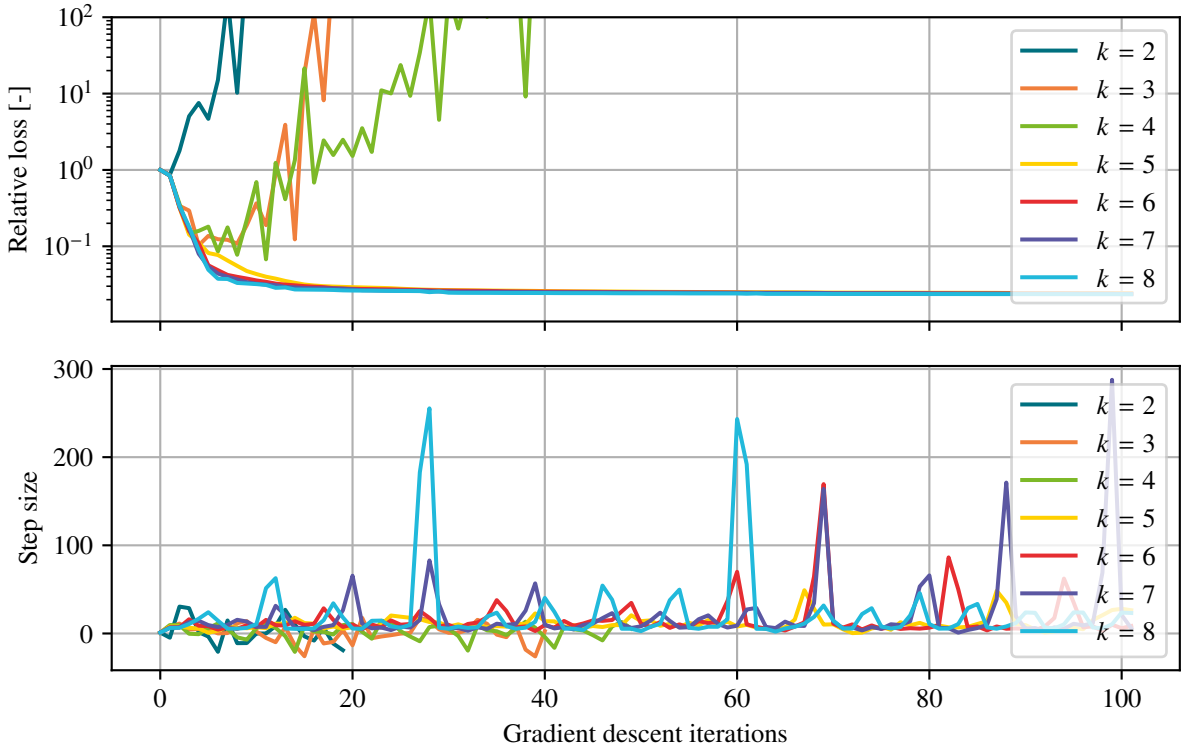
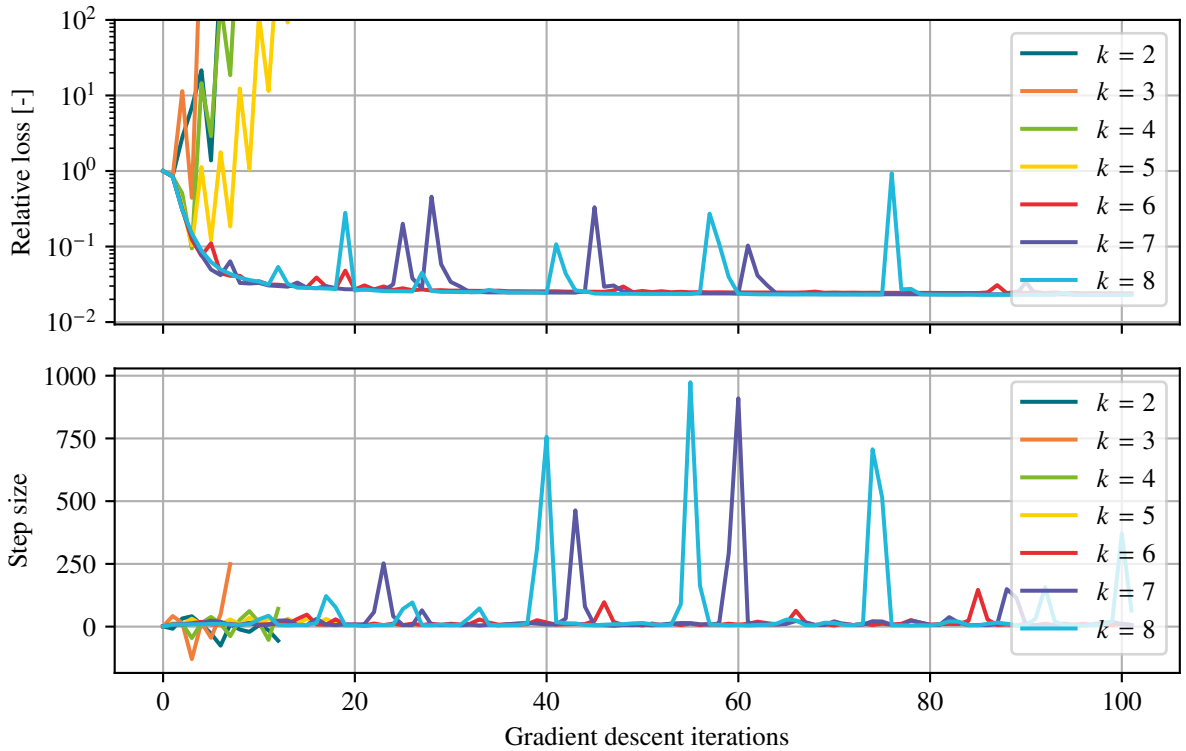


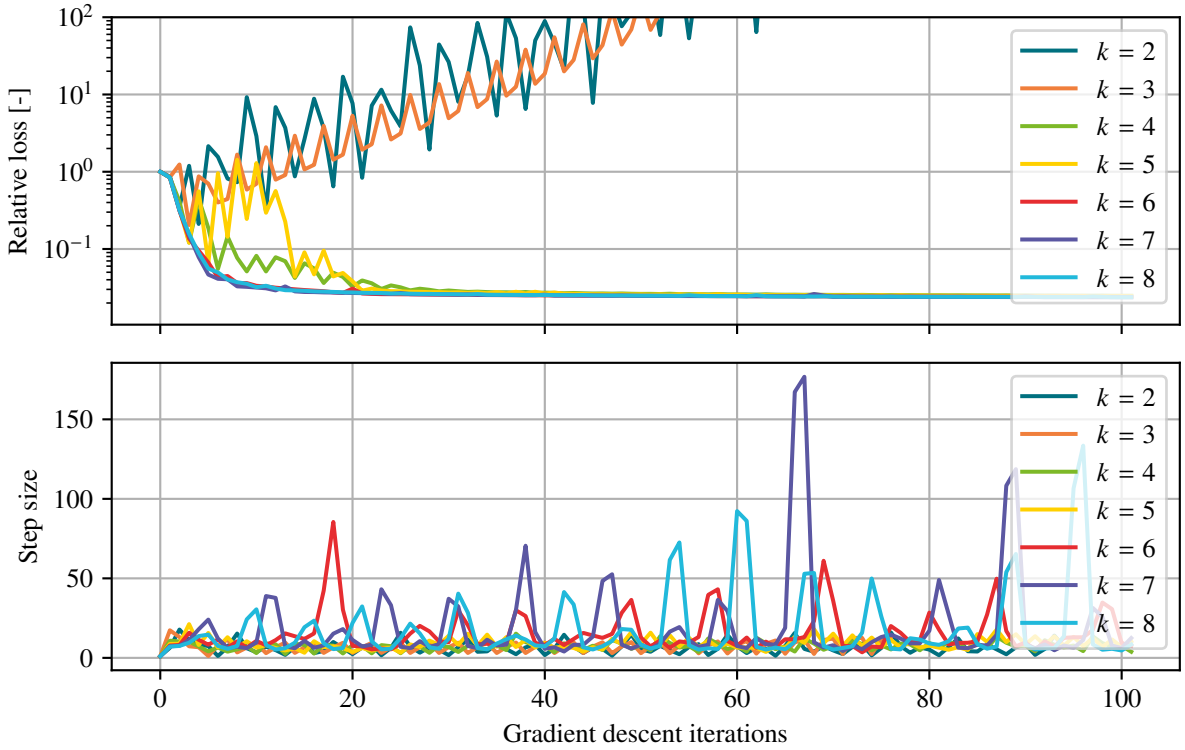
Figure 3.18: Convergence histories for exact inversion using GMRES(30) as the linear system solver, for the linearized inverse problem at 1 Hz in the Marmousi case, with 8 subdomains partitioned with METIS. The tolerance on the relative residual of the forward and adjoint fields is 10^{-6}. The optimization algorithm is the Barzilai-Borwein gradient descent with the short (3.22), long (3.23), and mean (3.24) step size choices. The initial step size is 1. The upper and lower graph are the relative loss and step sizes as a function of the gradient descent iteration, respectively. The relative loss is calculated with respect to its initial value.



(a) Short step Barzilai-Borwein (3.22).



(b) Long step Barzilai-Borwein (3.23).



(c) Mean step Barzilai-Borwein (3.24).

Figure 3.19: Convergence histories for sequential multi-step one-shot inversion using GMRES(k) as the linear system solver, for the linearized inverse problem at 1 Hz in the Marmousi case, with 8 subdomains partitioned with METIS. The optimization algorithm is the Barzilai-Borwein gradient descent with the short (3.22), long (3.23), and mean (3.24) step size choices. The initial step size is 1. The upper and lower graph are the relative loss and step sizes as a function of the gradient descent iteration, respectively. The relative loss is calculated with respect to its initial value.

of those observed in Figure 3.18 using exact inversion are observed. Interestingly, values of k that are in the higher end of the intermediate region, such as $k = 6$ display milder peaks than for $k = 7$ or $k = 8$. In the step size graph, it can be observed that for lower values of k ($k \leq 4$), the step size sometimes becomes negative; this phenomenon is likely caused by the estimated one-shot gradient increasing in magnitude despite going downhill in a convex landscape; these negative step sizes are likely the root cause of divergence for these values of k. Due to the weaker k-robustness and the presence of large spikes in the relative loss and step size for values of k that allow it to converge, the long step Barzilai-Borwein variant appears undesirable as an optimization algorithm for FWI using the one-shot paradigm.

Figures 3.19a and 3.19c display nicer trends with regards to FWI using the one-shot paradigm. Given that k is sufficiently high, both the short and mean step size variants do not show large spikes in the relative residual. Although the mean step size variant converges for $k = 4$ whereas the short step size does not, the convergence is somewhat highly nonmonotonous. In Figure 3.19a, the optimization diverges when $k \leq 4$, which can be attributed to negative step sizes being returned by the scheme. When it converges, the short step size appears to have a more predictable behavior

than the mean step size, with $k = 5$ showing smoother convergence. Interestingly, the step sizes of both variants have a tendency to be shorter as k decreases. Due to their higher k-robustness as well as relatively smooth convergence for high enough values of k, the short and mean Barzilai-Borwein variants appear more desirable as adaptive step size schemes.

Conclusions

The Barzilai-Borwein method, despite its simplicity, was found as an adequate optimization algorithm for the linearized inverse Helmholtz problem and fits relatively well with the one-shot paradigm, even when k is in a range of values which is well within the intermediate region (see Figure 3.14), implying that the method works satisfactorily even when the gradient estimation is largely inexact. Three variants of the schemes were tried: a short, long and mean step size, the latter of which has the property of being always positive.

The three step sizes were first tried in the context of inversion with a direct linear system solver. Although the short and mean step sizes showed similar good convergent behavior, the long step size exhibited significant unpredictable spikes in the relative loss.

The method was then tried in the context of one-shot optimization with GMRES(k) for values of k that are well within the intermediate region. Although the long step size was ruled out due to its lack of k-robustness and presence of spikes, the other two variants were found to be appropriate for FWI using one-shot. In particular, the mean step size displays a higher k-robustness due to never returning negative step sizes, whereas the short step size has much smoother convergence.

3.3.3 Perspectives

This section has investigated the Barzilai-Borwein method as a means to precondition the gradient and to appropriately scale it during the descent. Although step adaptivity alone is not sufficient to devise a fully adaptive one-shot optimization method, since k remains a parameter to be appropriately tuned, the method was found to be an adequate and simple means to precondition the gradient during descent with the one-shot paradigm provided that k is sufficient.

A logical way forward in exploring gradient preconditioning in a one-shot context would be to try to employ more complex quasi-Newton methods within the linearized inverse problem and to see how well they fare when used with inexact gradients. Although one could intuitively expect a tradeoff between the quality of the gradient estimation and the complexity of preconditioning method that can be used with it, some gains could perhaps be possible from lower order non-scalar approximations of the inverse Hessian (for example, by using a low order L-BFGS method).

Another possibility would be investigate the projection methods once again and to devise schemes to make them fit better in a one-shot optimization context. In addition, the nonlinear conjugate gradient variants remain to be thoroughly explored.

Finally, while this work remains limited to the scope of FWI through a succession of optimizations of linearized inverse problems (Gauss-Newton), it could be interesting to investigate one-shot optimization of the non-linearized inverse problem directly. While the need for globalization methods is not strong for the optimization of the convex and well-behaved misfit functional of the

linearized inverse problem, inversion of the full non-linearized inverse problem directly in one-shot would likely require a better attention to globalization methods and how to properly make them fit in a one-shot paradigm.

3.4 Adaptive k one-shot

As seen in the previous sections, the parameter k of the sequential multi-step one-shot algorithm has important effects on the behavior of the estimated gradient and on the optimization. For example, when k is too low, the estimated gradient is not able to follow the curvature sufficiently fast and therefore, the step size should be reduced. When k is sufficiently high, the estimated gradient displays a behavior which is similar to that of the exact gradient. When k is in a so-called intermediate region, the estimated gradient is sufficiently good for optimization but also shows surprising properties, such as robustness to higher step sizes. A comparison between the exact gradient and the one-shot estimated gradient seemed to indicate that, at least in the case considered, the estimated gradient follows the exact gradient during the descent, with some sort of momentum; the parameter k being likened to some strength with which the estimated gradient follows the exact one. The choice of k is a priori not obvious, since its optimal value highly depends on the conditioning of the forward and adjoint equations, and through this, the partition layout, number of subdomains (as seen in Section 3.2.1), the geometry of the model, among other things. Unlike the step size, the parameter k is much more akin to a hyperparameter of the one-shot inversion, and is more difficult to link with the conditions in which the optimization takes place.

In practice, it would be desirable that a non-suboptimal value of k permitting convergence using an adaptive step size scheme be automatically found during the optimization. In this section, an attempt is made at devising a simple strategy to adapt the value of k during the descent.

3.4.1 Negative step ramp-up

One of the simplest schemes to adapt k can be derived from the observations made in Section 3.3.2. As explained in that section, the linearized inverse Helmholtz problem misfit is a convex functional, therefore the short (3.22) and long (3.23) Barzilai-Borwein step sizes must necessarily be positive [9]. For a value of the multi-step one-shot parameter k to induce an estimated one-shot gradient that adequately models a convex functional landscape, it is therefore a necessary condition that the evaluation of either (3.22) or (3.23) using the one-shot estimated gradient yields a non-negative value.

Based on this necessary condition, we propose a simple "negative step ramp-up" scheme where the value of k is incremented whenever the condition

$$\langle \Delta g, \Delta \delta m \rangle < 0 \tag{3.25}$$

is met. At the beginning of the inversion, the value of k is set to an initial guess k^0 and occurrences of (3.25) cause a progressive sweep upwards during the optimization, until k reaches a value for which no occurrence of the condition happens. The k-adaptive sequential multi-step one-shot scheme is presented in Algorithm 11.

Algorithm 11 Abstract k-adaptive sequential multi-step one-shot optimization.

Require: `solve` and `optimize`
  $\delta u^0 = 0; \lambda^0 = 0; k = k^0$
  for $n = 1, 2, ...$ **do**
    $\delta u_0^n = \delta u^{n-1}; \lambda_0^n = \lambda^{n-1}$
    for $\ell = 1, 2, ..., k$ **do**
      $\delta u_\ell^n = \langle \ell$-th iteration of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U}\delta m^n \rangle$
    end for
    for $\ell = 1, 2, ..., k$ **do**
      $\lambda_\ell^n = \langle \ell$-th iteration of `solve` for $\tilde{A}^* \lambda = P_\Gamma^* (P_\Gamma \delta u_k^n - \delta d) \rangle$
    end for
    $\delta u^n = \delta u_k^n; \lambda^n = \lambda_k^n$
    $g^n = \texttt{grad}(\delta u^n, \lambda^n)$
    if $\langle g^n - g^{n-1}, \delta m^n - \delta m^{n-1} \rangle < 0$ **then**
      $k \leftarrow k + 1$
    end if
    $\delta m^{n+1} = \langle n$-th iteration of `optimize` $\rangle$
  end for

Although simple, the scheme presented in Algorithm 11 has some drawbacks. Firstly, no provision is taken for the downgrade of k; if the optimization process is such that the requirements on δu and λ to reach a sufficiently good estimated gradient loosen over the iterations, then this scheme is unable to adapt k downwards. A second drawback comes from the updating mechanism of k; if the first value of k for which (3.25) is never met is too high with respect to the initial guess k^0, then several potentially unproductive iterations are taken until adequate values of k are reached. Improving on this second aspect cannot be done with strategies such as aborting the step and recomputing the current gradient with an increased k, because in situations like the one described in Figure 3.17, the problem lies with the gradient of the previous iteration, as opposed to the gradient at the current iteration. Increasing k in a geometric manner could help, but the first value of k for which (3.25) is reached may then be missed. Nevertheless, this criterion and the settling values of k that the scheme produces are still to be investigated in this subsection.

Convergence histories

In one-shot optimization of the linearized inverse Helmholtz problem, we seek to observe how the relative loss and the value of k evolve along the optimization using Algorithm 11, starting from a value of $k^0 = 1$. The forward and adjoint solver is GMRES(k). For several values of the gradient descent step τ, 100 iterations of the fixed-step gradient descent are carried out on the linearized inverse Helmholtz problem at 1 Hz. The reference model is the Marmousi model, partitioned in 8 subdomains using METIS. The loss and the value of k are recorded at each iteration. Figure 3.20 shows the evolution of the relative loss and k as a function of the gradient iterations, for the different values of τ tested.

The upper graph of Figure 3.20 shows that when the step size is sufficiently low, the inversions are convergent. However, the relative loss diverges when the step is too high; interestingly, the

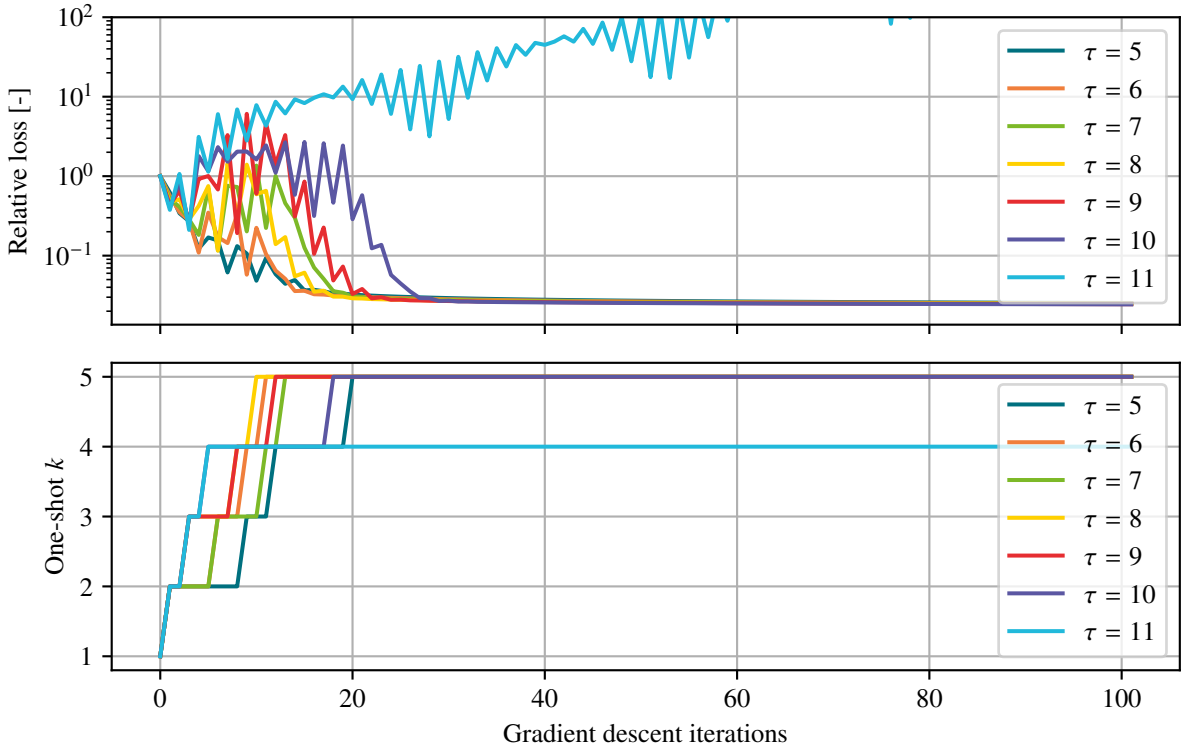


Figure 3.20: Convergence histories for the k-adaptive sequential multi-step one-shot inversion using GMRES(k) as the linear system solver using the negative step ramp-up scheme, for the linearized inverse problem at 1 Hz in the Marmousi case, with 8 subdomains partitioned with METIS. The optimization algorithm is the fixed-step gradient descent. The initial value of k is 1. The upper and lower graph are the relative loss and values of k as a function of the gradient descent iteration, respectively. The relative loss is calculated with respect to its initial value.

maximum allowable step of this scheme is higher than in the case of exact inversion, as seen in Figure 3.14. The figure also highlights one of the problems with the proposed scheme, in that as k increases, several bad gradient descent steps are taken before reaching a value that allows convergence, causing severe oscillations in the relative loss curves. The lower graph reveals that for all convergent step sizes, the value of k stabilizes at $k = 5$; this observation implies that although $k = 5$ is well within the intermediate k region for this configuration (see Figure 3.14), this value of k alongside the converging step sizes are sufficient for (3.25) to never occur.

The existence of such a settling value of k for a given step size τ is a direct consequence of the convexity of the linearized inverse problem misfit. Indeed, when k tends to infinity or that the forward and adjoint problems are solved exactly, then the measured value of $\langle \Delta g, \Delta \delta m \rangle$ is *always* non-negative by definition. Moreover, it was established in Section 3.3.2 that when k is sufficiently low, the measured value of $\langle \Delta g, \Delta \delta m \rangle$ may be negative. Therefore, there must exist some lower bound (let us call it k_a) on the set of values of k for which $\langle \Delta g, \Delta \delta m \rangle$ is always non-negative. There must also exist some lower bound (let us call it k_b) on the set of values of k for which $\langle \Delta g, \Delta \delta m \rangle$ is always non-negative for all $k \geq k_b$. Since k_b is located at the limit between a gradient behavior which is known to be transient, k_b is definitely not in the higher k region. If Algorithm 11 is run

for an infinite amount of outer iterations in a convergent configuration, then k will settle on k_a. If it can be said that $k_a = k_b$ in general, then the respect of the condition (3.25) can thus be seen, in a sense, as a meaningful criterion on the quality of the one-shot estimated gradient.

3.4.2 Perspectives

A simple k-adaptivity scheme based on the idea that the Barzilai-Borwein step size should be positive was proposed. Although easy to implement, this scheme suffers from the inability to downgrade the value of k, and from the increase of k being too slow. In addition, several questions on its correctness remain. Given that the gradient descent step size induces convergence of the optimization if the forward and adjoint fields are calculated exactly, is it true that $k = k_a$ and this step size define a convergent configuration? In addition, this scheme makes the assumption that the inverse problem is linearized, and cannot directly be fit to the non-linearized inverse problem, due to the presence of concavities in the functional.

There is much room to improve on k-adaptivity in several directions. In the frame of the negative step ramp-up scheme, one may imagine backtracking strategies when (3.25) occurs, allowing re-evaluation using a more accurate gradient from the previous step. This strategy would allow increasing k by several units by step, thereby reducing the number of unproductive steps taken before reaching the settling value of k. Another possibility would be to modulate k with techniques akin to globalization methods.

Much of the perceived clumsiness of k-adaptivity stems from the fact that k is a hyperparameter with few links to the actual optimization variables, and therefore has to be controlled with heuristics. A more natural way to condition the termination of the inner loops of one-shot optimization should be on some kind of criterion on either the residuals of the forward and/or adjoint problems, or on the evolution of the gradient estimations as the forward/adjoint iterations advance. Finally, it would be interesting to investigate if any parallels can be drawn between one-shot optimization with such kinds of criteria on the residuals and the work of van Leeuwen & Herrmann on penalty methods for PDE-constrained optimization in inverse problems [21].

3.5 Gauss-Newton one-shot

In Sections 3.3.2 and 3.4, an adaptive scheme for the step size and a heuristic for the update of the multi-step one-shot parameter k were explored. In this section, we combine those two methods in a single algorithm for the solution of the non-linearized inverse problem using Gauss-Newton, and benchmark it against L-BFGS(5) and Gauss-Newton-CG for solution of the non-linearized inverse problem.

As explained in Chapter 1, the optimization of a superquadratic functional is generally articulated in two steps. The first step consists of the calculation of a search direction as well as a first guess of an appropriate scaling; the second step consists of using a globalization strategy (*e.g.* a line search or a trust region method) to correct the scaling of the search direction such that the resulting step complies with some given criteria (*e.g.* Wolfe's, Armijo's, or Grippo-Lampariello-Lucidi conditions), generally ensuring global convergence. Until now, inversion was only considered

in the scope of only one linearized inverse problem, and given that the misfit functional of the linearized inverse problem is quadratic and convex, this led to most of the optimization schemes being well-behaved even in the absence of globalization strategy. In order to do the inversion of the non-linearized inverse problem while keeping the desirable optimization behaviors of the linearized inverse problem misfit, we consider here the Gauss-Newton method to find a search direction as well as its appropriate scale: a linearized inverse problem one-shot optimization scheme is equipped with a stopping criterion to find a search direction and its initial scaling, then a globalization strategy is applied on this search direction.

Gauss-Newton one-shot search direction. Due to the desirable convergence properties of the Barzilai-Borwein short step size and the better k-robustness of the mean step size, the following step size is chosen

$$p^n = \begin{cases} -\frac{||\Delta\delta m||}{||\Delta g||} g^n & \text{if } \langle \Delta g, \Delta\delta m \rangle < 0, \\ -\frac{\langle \Delta g, \Delta\delta m \rangle}{\langle \Delta g, \Delta g \rangle} g^n & \text{otherwise,} \end{cases} \tag{3.26}$$

alongside the use of the negative step ramp-up adaptive k scheme. The resulting method is shown in Algorithm 12.

Algorithm 12 Abstract k and step adaptive sequential multi-step one-shot optimization.

Require: `solve` and `optimize`
  $\delta u^0 = 0; \lambda^0 = 0; k = k^0$
  for $n = 1, 2, \ldots$ **do**
    $\delta u_0^n = \delta u^{n-1}; \lambda_0^n = \lambda^{n-1}$
    for $\ell = 1, 2, \ldots, k$ **do**
      $\delta u_\ell^n = \langle \ell$-th iteration of `solve` for $\tilde{A}\delta u = -\omega^2 \tilde{U}\delta m^n \rangle$
    end for
    for $\ell = 1, 2, \ldots, k$ **do**
      $\lambda_\ell^n = \langle \ell$-th iteration of `solve` for $\tilde{A}^*\lambda = P_\Gamma^*(P_\Gamma \delta u_k^n - \delta d) \rangle$
    end for
    $\delta u^n = \delta u_k^n; \lambda^n = \lambda_k^n$
    $g^n = \texttt{grad}(\delta u^n, \lambda^n)$
    if $\langle g^n - g^{n-1}, \delta m^n - \delta m^{n-1} \rangle < 0$ **then**
      $k \leftarrow k + 1$
      $p^n = -\frac{||\Delta\delta m||}{||\Delta g||} g^n$
    else
      $p^n = -\frac{\langle \Delta g, \Delta\delta m \rangle}{\langle \Delta g, \Delta g \rangle} g^n$
    end if
    $\delta m^{n+1} = \delta m^n + p^n$
  end for

The stopping criterion of the outer loop in Algorithm 12 was, until now, a fixed number of iterations. In the frame of one-shot optimization of a succession of linearized inverse problem, a more relevant stopping criterion can be chosen. The criterion considered here is one on the relative

norm of the one-shot estimated gradient, namely

$$\frac{||g^n||}{||\nabla_m L(m)||} < 0.5 \quad \text{or} \quad n \geq 10. \tag{3.27}$$

When solving the Newton equation of a quadratic functional, the residual is analogous to the negative gradient, as explained in Section 3.3.1, and therefore (3.27) is equivalent to a criterion on the relative residual, except that the residual is calculated in a one-shot fashion.

Globalization scheme. Armijo's backtracking line search is employed on the resulting search direction to ensure that the value of the misfit reduces after the step. Let δm be the step direction resulting from Algorithm 12, the line search is shown in Algorithm 4 [24].

Since Algorithm 12 is subject to direct comparison with other methods, the choice has been made to limit the scope of the one-shot optimization to the linearized inverse problem only to make the comparison fairer. Therefore, once a step direction δm has been computed in a one-shot fashion, the line search algorithm evaluates the non-linearized inverse problem misfit exactly. Likewise, the background fields of the linearization process are calculated exactly. In practice, one could imagine a scheme that is fully one-shot, even at the non-linearized level. In the context of Armijo's backtracking line search, for example, only a limited amount of iterations of a solution algorithm could be employed to calculate the waveform field needed to evaluate $L(m + \alpha\delta m)$, and by doing so, the background field of the next linearized inverse problem is already known.

3.5.1 Frequency sweep

In the previous section, Algorithm 12 was benchmarked against other algorithms for one given frequency. We aim to obtain higher fidelity images obtained with this algorithm by imaging on multiple frequencies. One approach consists of building a misfit as the sum of the misfit for all the instances of the inverse problem at different frequencies, as explained in Chapter 1; the resulting gradient is then used to optimize this global multi-frequency misfit. Another, more sequential approach consists of defining an ordered set of frequencies of interest, and then perform FWI on each of those frequencies successively; the squared slowness field m resulting from the FWI is used as initial guess for the next frequency. We shall refer to this second kind of approach as a frequency sweep.

In a practical setting, a frequency sweep requires a stopping criterion to be imposed on the inversion convergence, so as to allow the sweep to progress to the next frequency. For the following experiments, the stopping criterion is the relative misfit falling below 3×10^{-3} downwards

$$\frac{L(m_k)}{L(m_0)} < 3 \times 10^{-3}, \tag{3.28}$$

or stalling of the misfit; stalling is characterized by 3 successive misfits with a relative difference less than 10^{-2}. The frequencies of the sweep are selected on the basis of the scale of the problem, layout of the receivers and are also a function of the slowness field. In the following experiments, the frequencies are chosen within the characteristic range of frequencies given in Section 2.2, and

are spaced such that they meet the strategy of Sirgue and Pratt [33]. For each reference model, a minimum incidence aperture $\gamma_{\min}$ is calculated using

$$\gamma_{\min} = \frac{1}{\sqrt{1 + R_{\max}^2}}, \tag{3.29}$$

where $R_{\max}$ is the ratio between the maximal half offset $h_{\max}$ between a source and an emitter, and the depth of lowest scattering layer z – that is

$$R_{\max} = \frac{h_{\max}}{z}. \tag{3.30}$$

Starting from an initial frequency guess f_0 located at the lower end of the frequency ranges identified in Section 2.2, the set of frequencies is generated by the series

$$f_{n+1} = \frac{f_n}{\gamma_{\min}}. \tag{3.31}$$

Such a sweep is performed for all three reference models of this study: the Marmousi, 2004 BP and the T-shaped reflectors case. Among the values of interest during the sweep are the evolution of the relative misfit, of the value of k settled on by the update strategy (see Section 3.4), and the L^2 error between the current iterate of the model m and the projected reference model m_{ref}. The L^2 error on the model is defined as

$$E(m) = \sqrt{\int_\alpha \left(m(x) - m_{\text{ref}}(x)\right)^2 dx}, \tag{3.32}$$

where α is the region of the model space subject to optimization (see Section 2.2.4). The system resolution algorithm is GMRES(k). In each case, the domain is subdivided into 8 subdomains with METIS. For reference, these inversions will also be carried out using L-BFGS(5) and Gauss-Newton-CG, both using an Armijo line search; using those, the tolerance on the exact solution of the forward and adjoint fields is a relative decrease in residual of 10^{-4}.

Marmousi case

The Marmousi case is imaged using 120 sources and receivers laid out as specified in Section 2.2.4. The lower bound of the sweep f_0 is set to 1 Hz, as this frequency has been extensively studied in the various experiments of this chapter. The minimum incidence aperture $\gamma_{\min}$ of this configuration is given by

$$\gamma_{\min}^{-1} \simeq \sqrt{1 + \left(\frac{50 \times 9.24}{2 \times 52 \times 3.48}\right)^2} \simeq 1.6. \tag{3.33}$$

The image is continued until the sweep reaches the maximum frequency identified for the Marmousi case, as identified in Section 2.2. The set of frequencies selected as part of the sweep is therefore given by

$$\{1, 1.6, 2.5, 4, 6.5, 10.4\}\ \text{Hz}. \tag{3.34}$$

Figure 3.21 shows the squared wave slowness field that results from the inversion sweep under the specified parameters and frequency set, using the Gauss-Newton-OS algorithm. This result was obtained after 2702 preconditioner applications and the L^2 error E was decreased by 59.34%.

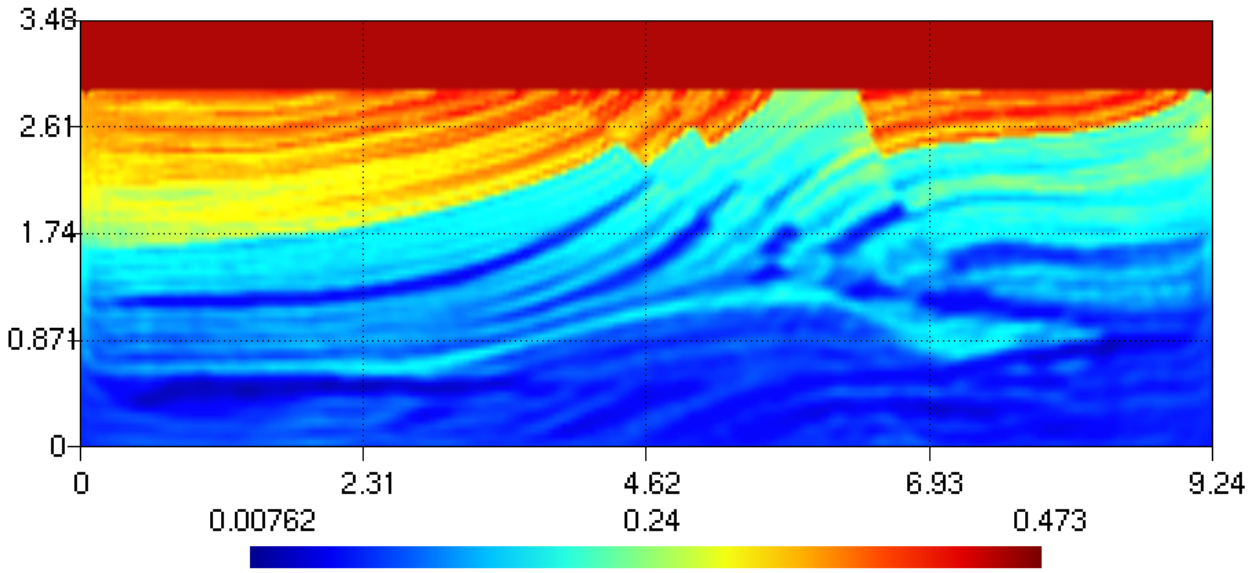


Figure 3.21: Squared wave slowness model that results from the inversion sweep at frequencies $\{1, 1.6, 2.5, 4, 6.5, 10.4\}$ Hz using Gauss-Newton-OS and the parameters described in Section 3.5.1 for the Marmousi case. The domain is partitioned into 8 subdomains with METIS. This result is obtained after 2702 preconditioner applications. The distances are in kilometers, and the squared wave slowness field is expressed in $\mathrm{s}^2\,\mathrm{km}^{-2}$.

A qualitative look at Figure 3.21, in comparison to the reference model (Figure 2.3) indicates that frequency sweep inversion using the Gauss-Newton-OS algorithm yielded a relatively accurate and high fidelity image. In particular, the inversion process could overcome the main intended challenges of the Marmousi model, as it was able to image despite the high slowness gradients and the discontinuities. The contrasts are better captured in the more shallow regions than in the deeper regions. Notwithstanding the actual accuracy of the field, such a result is likely to offer sufficient underground structural insight to a field engineer.

The evolution of $E(m)$ and of k as a function of preconditioner applications is shown in Figure 3.22, and the total number of preconditioner applications, system solves, gradient evaluations and assemblies is shown in Table 3.2. The figure highlights that all three methods were effective at reducing the model error and to perform FWI. Between the three methods considered, the L-BFGS(5) method is the most effective at reducing the model error. Comparing Gauss-Newton-OS and Gauss-Newton-CG shows that in the case and criteria considered, performing the inversion with one-shot within the linearized inverse problem is more cost effective. The lower graph shows that the one-shot k may increase as iterations and frequency advances, but ends up stagnating; such a behavior does not exclude the hypothesis that the ideal k does not follow a monotonous behavior along the Gauss-Newton iterations and frequencies. A strategy to downgrade k is therefore valuable. Table 3.2 indicates that, although an iteration of Gauss-Newton-CG requires on average

less evaluations of the gradient, the lower cost of gradient evaluation in one-shot leads to overall gains. The L-BFGS(5) method requires the least gradient evaluations, but requires the most assemblies due iterating on the non-linearized inverse problem.

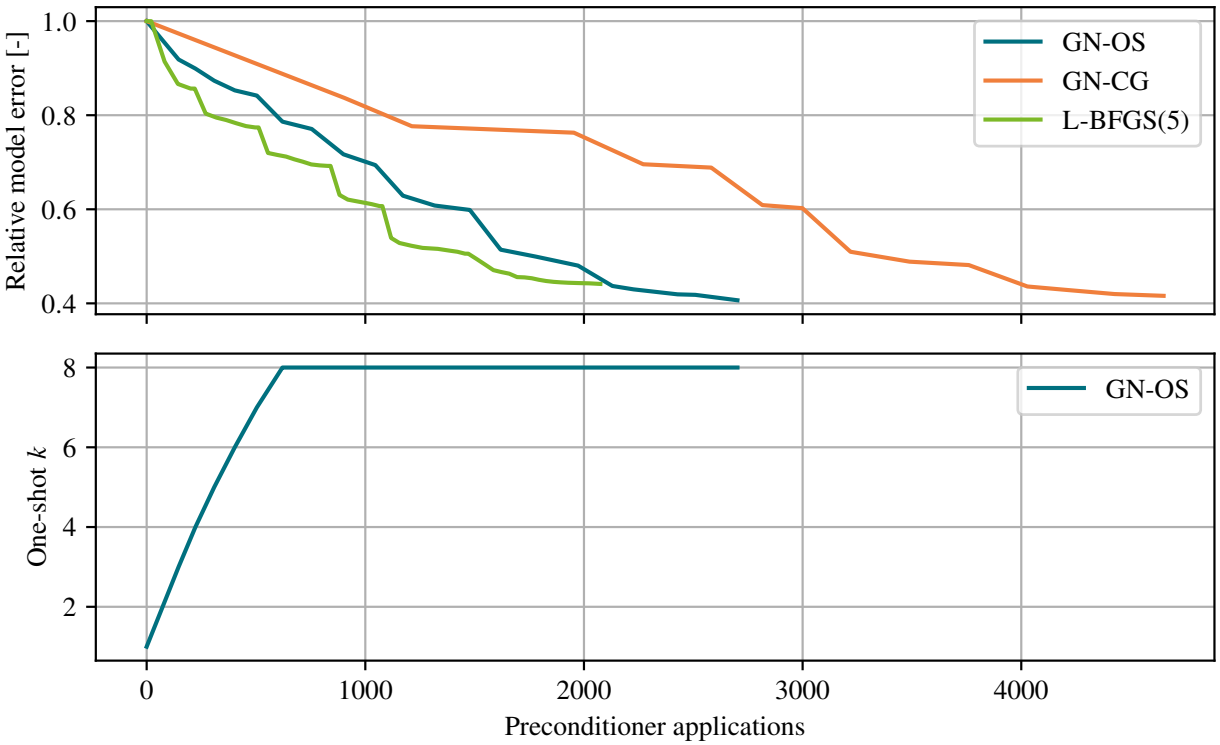


Figure 3.22: Evolution of the relative model error and of the one-shot parameter k resulting from the inversion sweep at frequencies $\{1, 1.6, 2.5, 4, 6.5, 10.4\}$ Hz using Gauss-Newton-OS, Gauss-Newton-CG and L-BFGS(5), with the parameters described in Section 3.5.1 for the Marmousi case. The domain is partitioned into 8 subdomains with METIS.

Method	Precond.	System solves	Grad. eval.	Assemblies
Gauss-Newton-OS	2702	369	164	41
Gauss-Newton-CG	4651	230	101	28
L-BFGS(5)	2075	167	53	114

Table 3.2: Number of preconditioner applications, system solves (including one-shot truncated solves), gradient evaluations and matrix assemblies needed to complete the inversion sweep at frequencies $\{1, 1.6, 2.5, 4, 6.5, 10.4\}$ Hz using Gauss-Newton-OS, Gauss-Newton-CG and L-BFGS(5), with the parameters described in Section 3.5.1 for the Marmousi case. The domain is partitioned into 8 subdomains with METIS.

2004 BP

The 2004 BP is imaged using 120 sources and receivers laid out as specified in Section 2.2.4. The lower bound of the sweep f_0 is set to 0.1 Hz, as it is at the lower end of the frequency range of

interest. The minimum incidence aperture $\gamma_{\min}$ of this configuration is given by

$$\gamma_{\min}^{-1} \simeq \sqrt{1 + \left(\frac{50 \times 67.4}{2 \times 52 \times 14.3}\right)^2} \simeq 2.4. \tag{3.35}$$

The set of frequencies selected to be part of the sweep is therefore

$$\{0.1, 0.24, 0.61, 1.51, 3.2\}\ \text{Hz}. \tag{3.36}$$

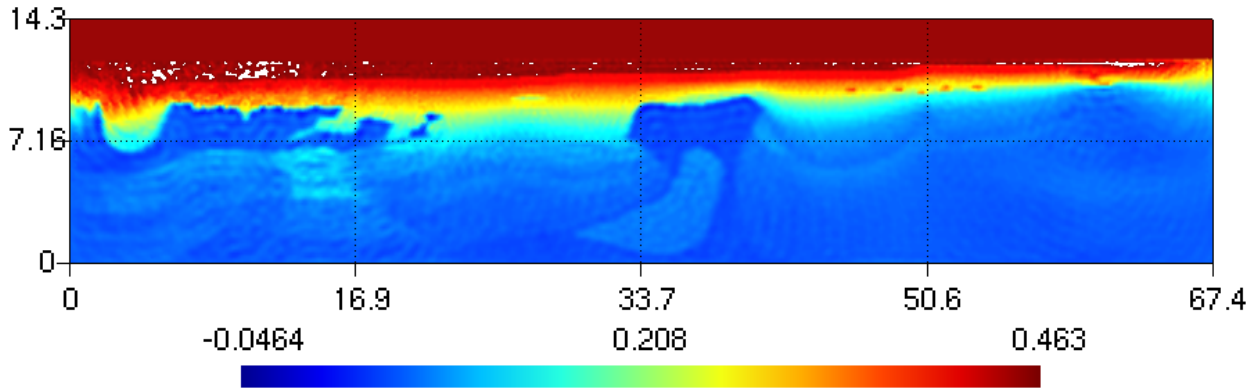


Figure 3.23: Squared wave slowness model that results from the inversion sweep at frequencies $\{0.1, 0.24, 0.61, 1.51, 3.2\}$ Hz using Gauss-Newton-OS and the parameters described in Section 3.5.1 for the 2004 BP case. The domain is partitioned into 8 subdomains with METIS. This result is obtained after 12304 preconditioner applications. The distances are in kilometers, and the squared wave slowness field is expressed in $\text{s}^2\,\text{km}^{-2}$. The initial k (k_0) is set to 15.

Figure 3.23 shows the squared wave slowness field that results from the sweep inversion obtained with Gauss-Newton-OS. The blank regions correspond to regions where the resulting field overshoots from the color scale range of the reference model. This image was obtained after 12304 preconditioner applications, and the model error has been reduced by 67.78%. Although the image suffers from noise and overshoots near the fixed water layer, a relatively high fidelity image was successfully obtained. In particular, the salt deposits were correctly imaged as well as the velocity anomalies underneath. In addition, the small scale deposits, on the right and at the center, were correctly recovered. The main defect of the resulting image is the noise, which was observed in all the images resulting from the three algorithms, the source of which is unclear.

Figure 3.24 shows the evolution of the relative model error and of the one-shot k along the frequency sweep. For the sweep inversion of this reference model and with the given parameters, L-BFGS(5) is once again able to reduce the relative model error the fastest; however, in this case, the Gauss-Newton-CG is more effective than Gauss-Newton-OS. Both Gauss-Newton-CG and Gauss-Newton-OS suffered from stalls at the later stages of the sweep. Similar observations as for the Marmousi case can be done for the evolution of k.

Table 3.3 shows the number of preconditioner applications, system solves, gradient evaluations and matrix assemblies needed to complete the inversion sweep for the three algorithms. The table

confirms the trends seen in the figure; once again, although L-BFGS(5) was the algorithm requiring the least gradient evaluations, it required the most assemblies.

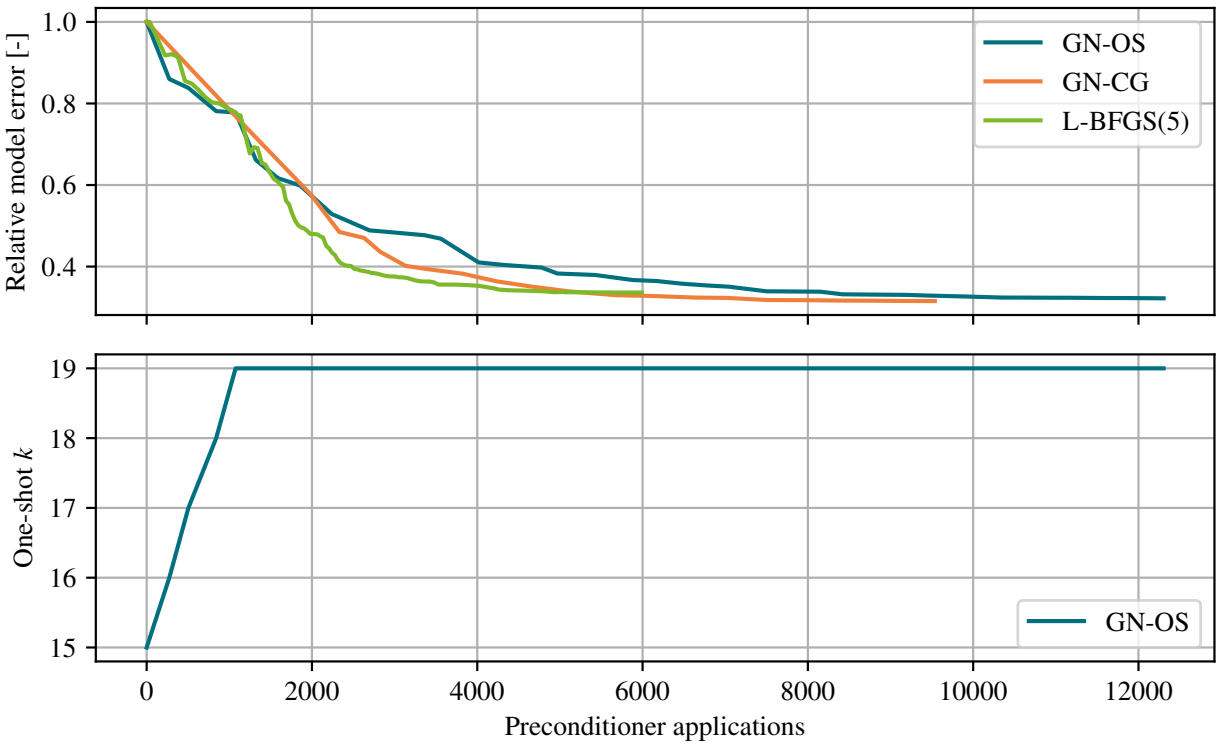


Figure 3.24: Evolution of the relative model error and of the one-shot parameter k resulting from the inversion sweep at frequencies $\{1, 1.6, 2.5, 4, 6.5, 10.4\}$ Hz using Gauss-Newton-OS, Gauss-Newton-CG and L-BFGS(5), with the parameters described in Section 3.5.1 for the 2004 BP case. The domain is partitioned into 8 subdomains with METIS.

Method	Precond.	System solves	Grad. eval.	Assemblies
Gauss-Newton-OS	12304	715	320	75
Gauss-Newton-CG	9536	360	158	44
L-BFGS(5)	5990	369	111	258

Table 3.3: Number of preconditioner applications, system solves (including one-shot truncated solves), gradient evaluations and matrix assemblies needed to complete the inversion sweep at frequencies $\{1, 1.6, 2.5, 4, 6.5, 10.4\}$ Hz using Gauss-Newton-OS, Gauss-Newton-CG and L-BFGS(5), with the parameters described in Section 3.5.1 for the 2004 BP case. The domain is partitioned into 8 subdomains with METIS.

T-shaped reflectors

The last sweep inversion is performed on the T-shaped reflectors reference model defined in Section 2.2. This model is imaged using 120 sources and receivers laid out as specified in

Section 2.2.4. The frequencies used for the sweep are not selected on the basis of the strategy of Sirgue and Pratt [33], but instead based on those used by [23], who introduced the case. The frequencies of the sweep are

$$\{100, 125, 150, 175, 200, 225, 250, 275, 300\}\ \text{Hz}. \tag{3.37}$$

Figure 3.25 shows the squared wave slowness field that results from the sweep inversion at the given frequencies and stopping criteria, obtained using the Gauss-Newton-OS algorithm. This image was obtained in 4090 preconditioner applications and the model error was decreased by 52.9%. Barring some overshoots and undershoots with respect to the reference model scale (also seen in the resulting image of Gauss-Newton-CG), highlighting difficulties near the very high slowness gradients, a relatively clear image of the T-shaped reflectors was retrieved. The region underneath the T-shaped reflectors is less well imaged, with irregularities in the bottom reflective layer.

Figure 3.26 shows the relative model error and the one-shot parameter k as a function of the number of preconditioner applications, for the three algorithms. In this particular imaging scenario, the Gauss-Newton-OS algorithm was the most efficient at decreasing the relative model error quickly, followed by Gauss-Newton-CG. Interestingly, in this imaging scenario, the relative model error induced by L-BFGS(5) increases after about 4000 preconditioner applications; this can probably be attributed to oversolving after a frequency sweep that was too fast, due to the stalling avoidance mechanism ending the inversion at a given frequency. In this imaging scenario, the one-shot k seems to steadily increase as the sweep progresses.

Table 3.4 lists the number of preconditioner applications, system solves, gradient evaluations and matrix assemblies needed to complete the inversion for the three algorithms. Gauss-Newton-OS is once again the method that requires the most gradient evaluations, although the lower cost of calculation lowers the overall cost of the inversion with respect to Gauss-Newton-CG, the former requiring less than half as many preconditioner applications than the latter. In the table, the L-BFGS(5) line is not meaningful due to the method failing to produce a correct inversion.

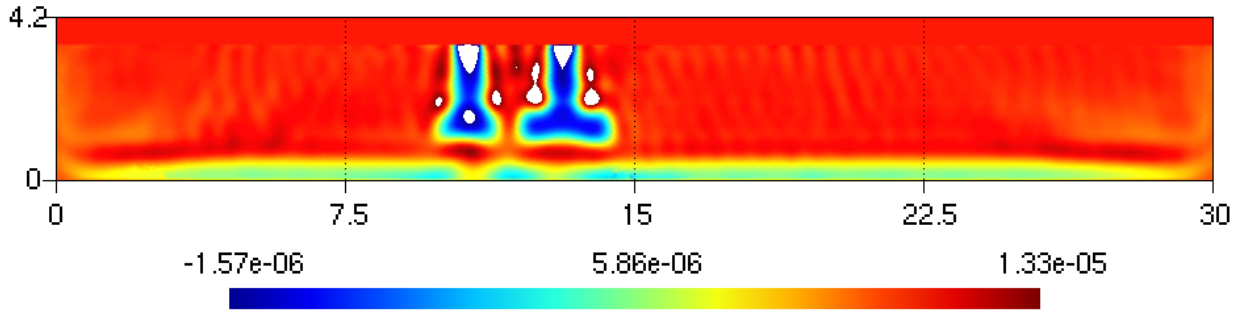


Figure 3.25: Squared wave slowness model that results from the inversion sweep at frequencies $\{100, 125, 150, 175, 200, 225, 250, 275, 300\}$ Hz using Gauss-Newton-OS and the parameters described in Section 3.5.1 for the T-shaped reflectors case. The domain is partitioned into 8 subdomains with METIS. This result is obtained after 4090 preconditioner applications. The distances are in meters, and the squared wave slowness field is expressed in $\text{s}^2\,\text{m}^{-2}$.

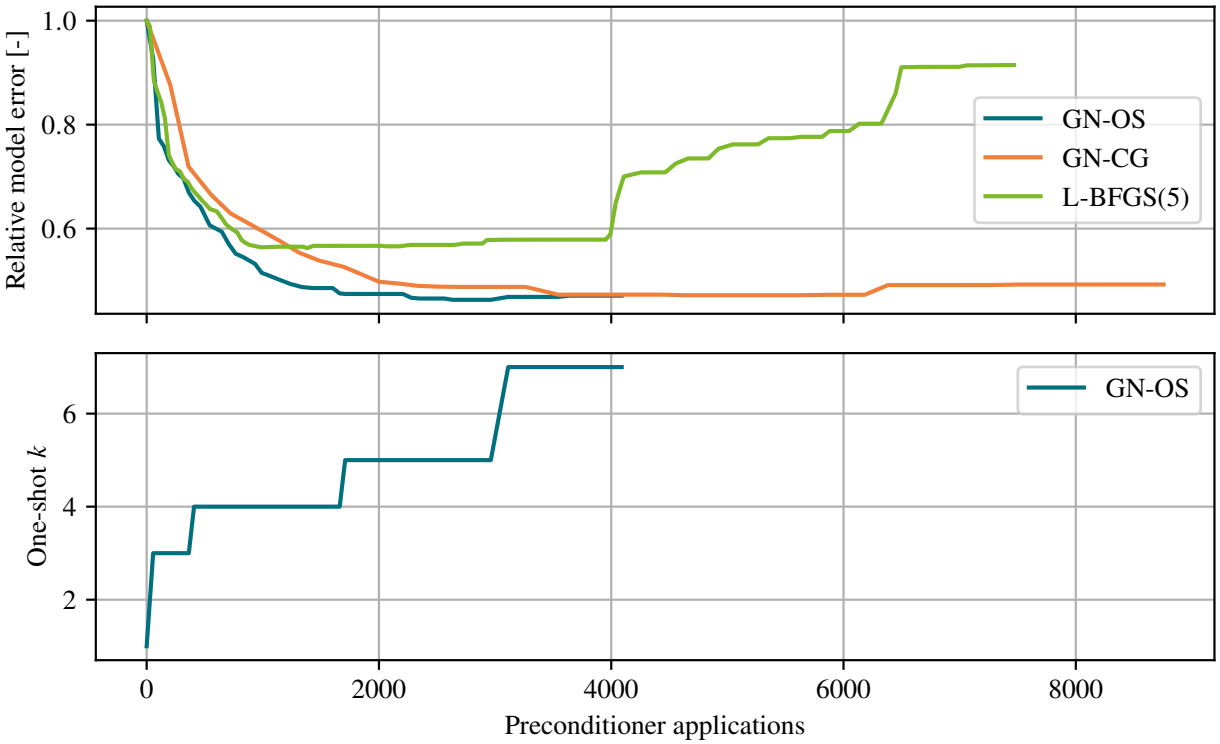


Figure 3.26: Evolution of the relative model error and of the one-shot parameter k resulting from the inversion sweep at frequencies $\{100, 125, 150, 175, 200, 225, 250, 275, 300\}$ Hz using Gauss-Newton-OS, Gauss-Newton-CG and L-BFGS(5), with the parameters described in Section 3.5.1 for the T-shaped reflectors case. The domain is partitioned into 8 subdomains with METIS.

Method	Precond.	System solves	Grad. eval.	Assemblies
Gauss-Newton-OS	4090	862	303	256
Gauss-Newton-CG	8756	650	202	246
L-BFGS(5)	7468	547	78	469

Table 3.4: Number of preconditioner applications, system solves (including one-shot truncated solves), gradient evaluations and matrix assemblies needed to complete the inversion sweep at frequencies $\{100, 125, 150, 175, 200, 225, 250, 275, 300\}$ Hz using Gauss-Newton-OS, Gauss-Newton-CG and L-BFGS(5), with the parameters described in Section 3.5.1 for the T-shaped reflectors case. The domain is partitioned into 8 subdomains with METIS.

Conclusions

The results of this section showed that Gauss-Newton-OS is not only a viable algorithm to obtain correct results, but that it also displays performances that are comparable to those of Gauss-Newton-CG and L-BFGS. Although which algorithm is the most effective depends ultimately on the imaging scenario and its parameters, Gauss-Newton-OS can sometimes exhibit better performance than Gauss-Newton-CG (Marmousi and T-reflectors cases) and sometimes outperform L-BFGS

(T-reflectors case). Comparing both Gauss-Newton methods in way the Newton equation is solved, although the one-shot variant steps in the direction of the gradient as opposed to a better direction, the lower cost of gradient evaluation makes the overall scheme cheaper in several cases.

Numerical experience when elaborating the sweeps showed that the performance of each algorithm with respect to each other is highly dependent on the stopping criteria chosen for switching to the next frequency, and for finishing a Gauss-Newton iteration, which makes the rigorous comparison a difficult endeavor. In addition, inversion is highly sensitive to the initial guesses and frequencies chosen for inversion. Also, in the context of this work, some shortcuts have been taken in the implementation of L-BFGS to make it more easily comparable to Gauss-Newton optimization; an ideal implementation of L-BFGS requires a globalization method that respects the Wolfe conditions, whereas a simple Armijo backtracking line search was used here (alongside a skip of the BFGS update when it would make the Hessian approximation indefinite). It is unclear whether enforcing Wolfe conditions would make the L-BFGS scheme more cost-effective however, due to the higher number of gradient evaluations needed.

Conclusions and perspectives

The purpose of this master's thesis was to numerically investigate the multi-step one-shot paradigm for full waveform inversion. More specifically, the goal was articulated around two axes. The first axis was to modify an existing FWI code so as to introduce the one-shot paradigm for the simplest possible configuration and, from this, to build an understanding of how optimization behaves under one-shot. The second axis was to build upon this experimental knowledge on the one-shot paradigm to, step by step, build a practical and adaptive one-shot FWI scheme by first introducing step size, then k-adaptivity into the method.

Before delving into one-shot methods, gathering and setting the general theoretical context in which FWI is performed was necessary. The mathematical background behind FWI was first recalled in Chapter 1. The first part of the chapter was about the Helmholtz problem and its discretization: how to go from the problem of acoustic wave propagation all the way to obtaining the discretized system ready to be practically solved on a computer. The logical next step was then to explain how these can be solved with a computer, in particular, with iterative methods; this section introduced a few iterative methods of historical or practical relevance, as well as the preconditioner employed to make the iterative resolutions possible. The last part of this chapter addressed the inverse problem in itself, from formally stating the inverse problem to the optimization methods that are typically employed for FWI.

From the theoretical background built in Chapter 1, a link to the practical world was required to frame the context in which the experiments take place. For this, the reference solver and the reference cases were introduced in Chapter 2.

The one-shot method was then investigated in Chapter 3. As a first step, the paradigm was tested in the context of gradient descent, with a fixed step, on the linearized inverse problem. What was found is that the one-shot paradigm not only is effective at reducing the cost of gradient descent for a given step size, but that in some cases, it can make the optimization convergence more robust against longer steps. This has prompted interest in the study of the evolution of the estimated gradient as a function of the parameter k, and it was found that the estimated gradient passes through a series of intermediate states that progressively tend to the exact gradient in both angle and magnitude. This observation led to the formulation of qualitative explanations to several of the behaviors previously witnessed.

Still in the inversion of the linearized inverse problem, and motivated by the search towards a more practical scheme, step-size adaptivity was introduced into the one-shot gradient descent in Section 3.3. A first reflection on the usage of projection methods to solve the Newton equation

lead to the conclusion that they are not a natural fit for the one-shot paradigm due to the need to evaluate the gradient at a non-iterate. Instead, quasi-Newton methods were found to be a better fit. Step size adaptivity was introduced using the Barzilai-Borwein method, and it has been found to be adequate for FWI, and robust against one-shot estimated gradients, as long as k has a sufficient value.

Based on the requirement that the Barzilai-Borwein step size be positive, a simple scheme to adapt k on the fly was devised in Section 3.4. Although working for simple cases, this scheme suffers from the inability to downgrade k, and from the slowness of the initial ramp-up, causing several bad steps to be taken.

Finally, the adaptive step and k schemes were put together into a novel Gauss-Newton scheme for the inversion of the full, non-linearized inverse problem in Section 3.5. This scheme was found to be comparable in performance to other optimization algorithms that are commonly used in FWI, and sometimes outperformed them.

Improvements and perspectives

The Barzilai-Borwein method is an interesting and satisfactorily simple scheme for gradient descent step size adaptivity. It would be interesting to investigate other more complex quasi-Newton methods to precondition the gradient in both scale and direction to improve descent efficiency. In addition, projection methods and methods akin to the nonlinear conjugate gradient should be explored in more detail.

One aspect of the multi-step one-shot methods that remains rather elusive, and perhaps the main hurdle is the hyperparameter k and how to adapt it such that it is not suboptimal for the given problem. Since k is a discrete parameter with few links to the actual optimization variables, knowing whether k is inadequate simply from the optimization variables is not obvious. In addition, the proposed scheme for k-adaptivity breaks in situations where the misfit functional is supposed to have concave regions. It would be interesting to explore alternative means of determining the halting of the iterative solves based on metrics meaningful within the optimization, such as the evolution of the gradient itself.

When it comes to the Gauss-Newton-OS scheme, a significant bottleneck is the calculation of the background field, which is inherent to the use of Gauss-Newton. A way to reduce this bottleneck would be to explore means of performing the one-shot optimization on the non-linearized inverse problem directly, and to integrate globalization into the scheme in a one-shot manner as well. A problem with this approach is the lack of convexity guarantees which breaks the assumptions underlying the Barzilai-Borwein method and the k-adaptive scheme. These would therefore need to be rethought entirely.

Bibliography

[1] X. Adriaens, "Inner product preconditioned optimization methods for full waveform inversion," Ph.D. dissertation, University of Liège, Dec. 2022.

[2] X. Adriaens, L. Métivier, and C. Geuzaine, "Inner product preconditioned trust-region methods for frequency-domain full waveform inversion," *Journal of Computational Physics*, vol. 493, p. 112 469, 2023.

[3] S. Balay et al., "PETSc Users Manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.6, 2015.

[4] S. Balay et al., *PETSc Web page*, `http://www.mcs.anl.gov/petsc`, 2015.

[5] J. Barzilai and J. M. Borwein, "Two-Point Step Size Gradient Methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[6] F. Billette and S. Brandsberg-Dahl, "The 2004 BP Velocity Benchmark," cp-1-00513, 2005.

[7] M. Bonazzoli, H. Haddar, and T. A. Vu, "Convergence analysis of multi-step one-shot methods for linear inverse problems," *arXiv preprint arXiv.2207.10372*, Jul. 2022.

[8] A. M. Bradley, *PDE-constrained optimization and the adjoint method*, 2024.

[9] Y.-H. Dai, M. Al-Baali, and X. Yang, "A Positive Barzilai–Borwein-Like Stepsize and an Extension for Symmetric Linear Systems," in *Numerical Analysis and Optimization*, M. Al-Baali, L. Grandinetti, and A. Purnama, Eds., Cham: Springer International Publishing, 2015, pp. 59–75.

[10] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2015.

[11] O. G. Ernst and M. J. Gander, "Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods," in *Numerical Analysis of Multiscale Problems* (Lecture Notes in Computational Science and Engineering), I. G. Graham, T. Y. Hou, O. Lakkis, and R. Scheichl, Eds., Lecture Notes in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 83, pp. 325–363.

[12] R. Fletcher, "On the Barzilai-Borwein Method," in *Optimization and Control with Applications*, L. Qi, K. Teo, and X. Yang, Eds., Boston, MA: Springer US, 2005, pp. 235–256.

[13] H. Fu, M. Ma, and B. Han, "An accelerated proximal gradient algorithm for source-independent waveform inversion," *Journal of Applied Geophysics*, vol. 177, p. 104 030, 2020.

[14] T. Gabriel, "Acceleration of Frequency-Domain Full Wave Inversion through Krylov Reuse," M.S. thesis, University of Liège, Liège, Belgium, 2023.

[15] S. Gong, I. G. Graham, and E. A. Spence, "Convergence of Restricted Additive Schwarz with impedance transmission conditions for discretised Helmholtz problems," *arXiv preprint arXiv.2110.14495*, Jun. 2022.

[16] L. Grippo, F. Lampariello, and S. Lucidi, "A Nonmonotone Line Search Technique for Newton's Method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, Aug. 1986.

[17] L. Guasch, O. Calderón Agudo, M.-X. Tang, P. Nachev, and M. Warner, "Full-waveform inversion imaging of the human brain," *npj Digital Medicine*, vol. 3, no. 1, pp. 1–12, Mar. 2020.

[18] G. Karypis and V. Kumar, *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, Sep. 1998.

[19] Y. Kida, H. Mikada, T.-n. Goto, and J. Takekawa, "Application of the full-waveform inversion techniques to the estimation of the sound velocity structure in the ocean," in *2012 Oceans*, Oct. 2012, pp. 1–7.

[20] A. Klotzsche, J. van der Kruk, G. A. Meles, J. Doetsch, H. Maurer, and N. Linde, "Full-waveform inversion of crosshole ground penetrating radar data to characterize a gravel aquifer close to the Thur River, Switzerland," in *Proceedings of the XIII International Conference on Ground Penetrating Radar*, Jun. 2010, pp. 1–5.

[21] T. van Leeuwen and F. J. Herrmann, "A penalty method for PDE-constrained optimization in inverse problems," *Inverse Problems*, vol. 32, no. 1, p. 015 007, Dec. 2015.

[22] B. Martin and A. Sior, *Distributed FWI*, `https://gitlab.onelab.info/boris-martin/distributed_fwi`, branch `master`, accessed 2025-05-14 14:57, 2025.

[23] L. Métivier, R. Brossier, S. Operto, and J. Virieux, "Full Waveform Inversion and the Truncated Newton Method," *SIAM Review*, vol. 59, no. 1, pp. 153–195, Jan. 2017.

[24] J. Nocedal and S. J. Wright, *Numerical optimization* (Springer series in operations research and financial engineering), 2nd ed. New York: Springer, 2006.

[25] S. Operto et al., "Efficient 3-D frequency-domain mono-parameter full-waveform inversion of ocean-bottom cable data: application to Valhall in the visco-acoustic vertical transverse isotropic approximation," *Geophysical Journal International*, vol. 202, no. 2, pp. 1362–1391, Aug. 2015.

[26] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, "Recycling Krylov Subspaces for Sequences of Linear Systems," *SIAM Journal on Scientific Computing*, vol. 28, no. 5, pp. 1651–1674, Jan. 2006.

[27] M. Raydan, "On the Barzilai and Borwein choice of steplength for the gradient method," *IMA Journal of Numerical Analysis*, vol. 13, no. 3, pp. 321–326, Jul. 1993.

[28] M. Raydan, "The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 26–33, Feb. 1997.

[29] A. Royer, "Efficient finite element methods for solving high-frequency time-harmonic acoustic wave problems in heterogeneous media," Ph.D. Thesis, University of Liège, 2023.

[30] S. Saab, S. Phoha, M. Zhu, and A. Ray, "An adaptive polyak heavy-ball method," *Machine Learning*, vol. 111, no. 9, pp. 3245–3277, Sep. 2022.

[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, 2003.

[32] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, Jul. 1986.

[33] L. Sirgue and R. G. Pratt, "Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies," *GEOPHYSICS*, vol. 69, no. 1, pp. 231–248, Jan. 2004.

[34] P.-H. Tournier, P. Jolivet, V. Dolean, H. S. Aghamiry, S. Operto, and S. Riffo, "Three-dimensional finite-difference & finite-element frequency-domain wave simulation with multi-level optimized additive Schwarz domain-decomposition preconditioner: A tool for FWI of sparse node datasets," *arXiv preprint arXiv.2110.15113*, Oct. 2021.

[35] R. Versteeg, "The Marmousi experience: Velocity model determination on a synthetic complex data set," *The Leading Edge*, vol. 13, no. 9, pp. 927–936, 1994.

[36] J. Virieux, A. Asnaashari, R. Brossier, L. Métivier, A. Ribodetti, and W. Zhou, "An introduction to full waveform inversion," in *Encyclopedia of Exploration Geophysics*. 2017, ch. 6, R1–1-R1–40.

[37] T.-A. Vu, "Méthodes d'inversion de type one-shot et décomposition de domaine," Ph.D. dissertation, Institut Polytechnique de Paris, Jul. 2024.

[38] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*, 7th ed. Amsterdam: Elsevier, Butterworth-Heinemann, 2013.

Appendix A: Initial models

This appendix contains the initial models that are used for the inversion experiments. The initial models used for Marmousi, 2004 BP and T-shaped reflectors reference cases are shown in Figures 3.27, 3.28 and 3.29 respectively.

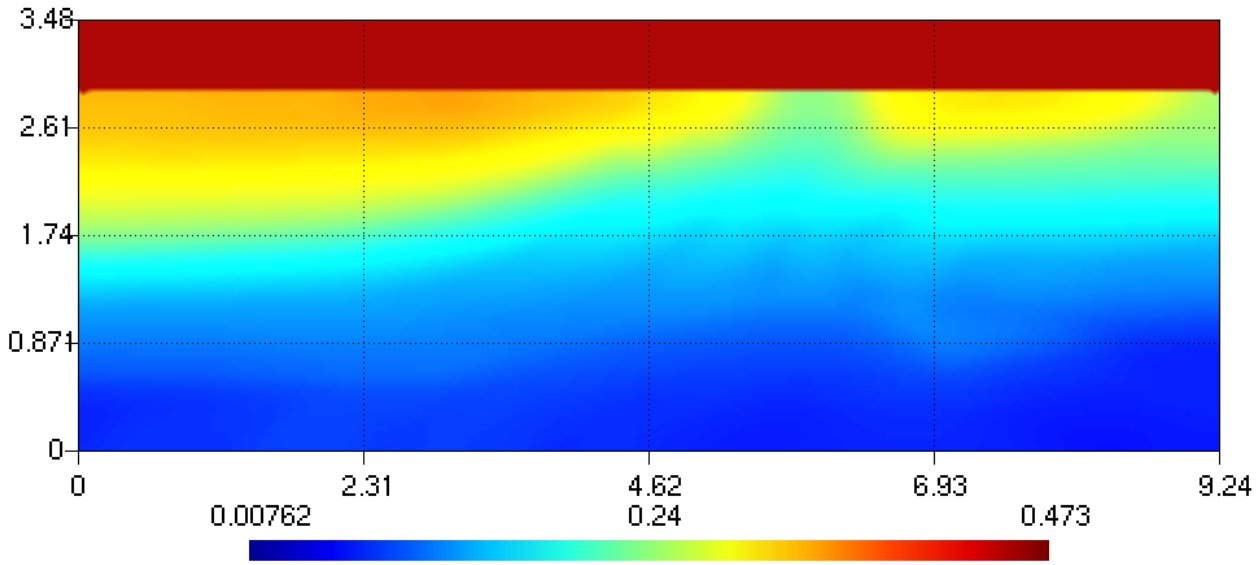


Figure 3.27: The initial model used for inversion experiments using the Marmousi squared wave slowness model [35]. The distances are expressed in kilometers, and the squared wave slowness field is expressed in $s^2\,km^{-2}$. A water layer has been added on top and is considered known during inversion. This initial model was obtained from the reference using a smoothing factor of $0.1\,m^2$.

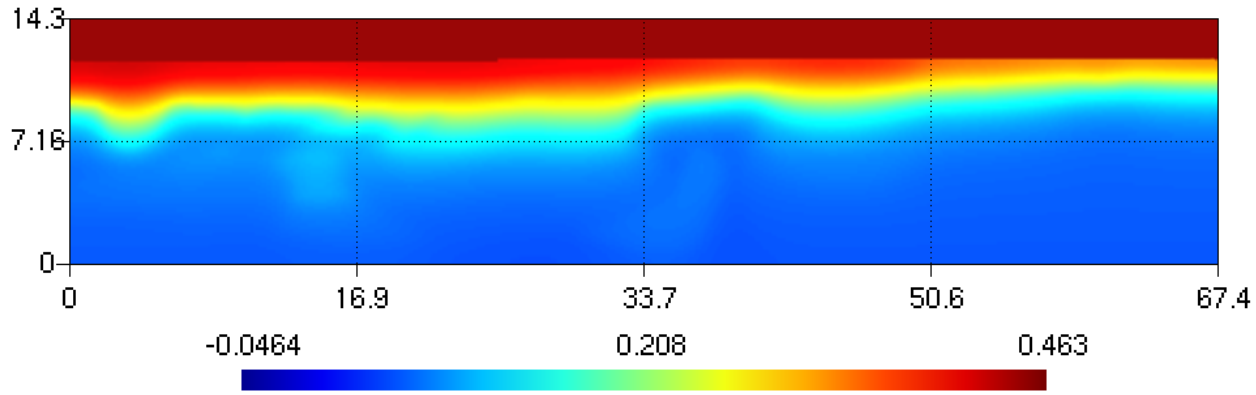


Figure 3.28: The initial model used for inversion experiments using the 2004 BP squared wave slowness model [6]. The distances are expressed in kilometers, and the squared wave slowness field is expressed in $s^2\,km^{-2}$. A water layer has been added on top and is considered known during inversion. This initial model was obtained from the reference using a smoothing factor of $1.0\,m^2$.

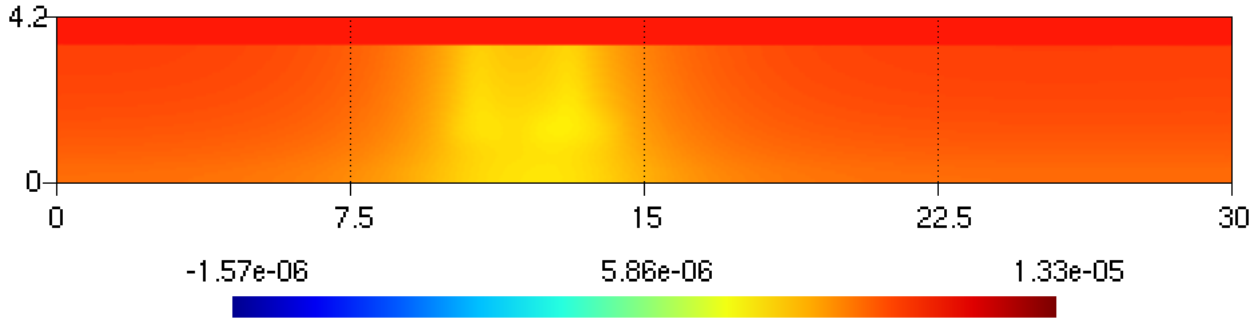


Figure 3.29: The T-shaped reflector squared wave slowness model [1], [23]. The distances are expressed in meters, and the squared wave slowness field is expressed in $s^2\,m^{-2}$. A soil layer has been added on top and is considered known during inversion. The minimum and maximum values of the slowness squared are rendered inaccurate by undershoots and overshoots at projection. This initial model was obtained from the reference using a smoothing factor of $5.0\,m^2$.