Université de Liège

Faculté des Sciences Appliquées

Collège de doctorat en Électricité, Électronique et Informatique

Inner product preconditioned optimization methods for full waveform inversion

Doctoral dissertation presented by

Xavier Adriaens

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (PhD) in Engineering Sciences

December 2022

Thesis committee:

Prof. Maarten Arnst (Université de Liège)

Prof. Christophe Geuzaine (Université de Liège), Supervisor

Dr. Ludovic Métivier (CNRS - Université Grenoble Alpes)

Prof. Frédéric Nguyen (Université de Liège)

Prof. Tristan van Leeuwen (Universiteit Utrecht)

Prof. Wim Vanroose (Universiteit Antwerpen)

Author contact information:

Xavier Adriaens

ACE - Applied and Computational Electromagnetics,
Dept. of Electrical Engineering and Computer Science,
Université de Liège

Montefiore Institute B28,
Quartier Polytech 1,
Allée de la Découverte 10,
4000 Liège, Belgium

Email: xavier.adriaens@uliege.be

Funding:

This thesis was funded by the Fonds de la Recherche Scientifique (F.R.S.-FNRS).

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.

 This version was generated on December 20, 2022

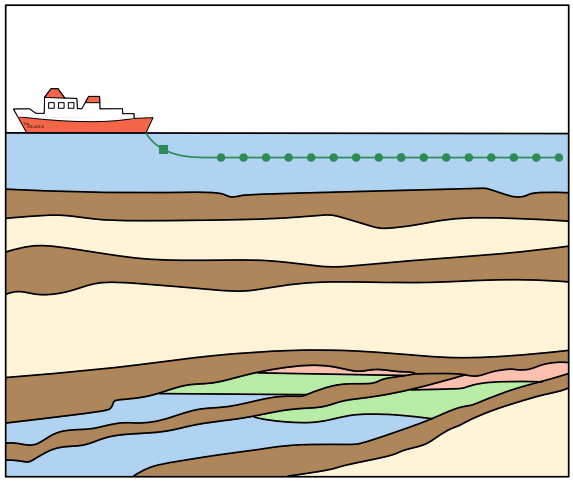
Contents

Introduction

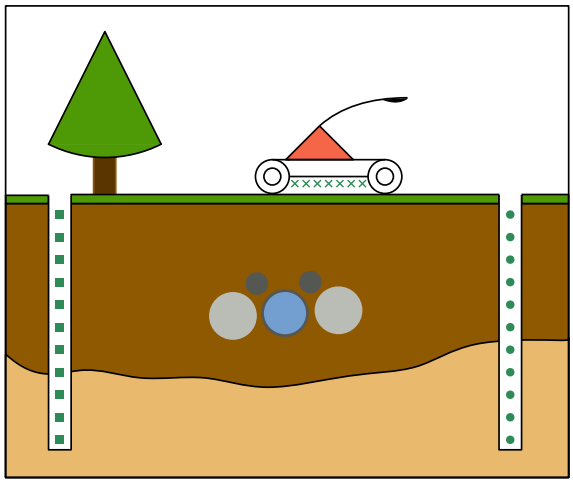
Context and motivations

The interpretation of wave scattering data has led, and continues to lead, to a wide range of scientific discoveries and practical engineering applications in domains as varied as materials science, biomedical imaging, geophysics, atmospheric science, oceanography, plasma physics, archaeology, astrophysics or quantum information. A special case is tomography: the process of reconstructing the volume properties of a sample from scattered wave measurements collected on its boundary. Under various assumptions linked to the physical nature of the waves (acoustic, elastic, electromagnetic, *etc.*) and the properties of the sample, tomography mathematically reduces to the solution of an inverse wave scattering problem. When the scattering is weak and receivers can be placed around the sample, explicit analytical solutions can be obtained using filtered back projection, a solution procedure based on the inverse Radon transform [125]. If the scattering is stronger or if sources and receivers can only be placed on a limited part of the boundary, iterative reconstruction must be used instead. Basically any iterative reconstruction method consists in minimizing the mismatch between the measured wave scattering data and some simulated data, obtained through a forward wave propagation model of the scattering phenomenon [33, 29, 40, 141]. Full waveform inversion (also called waveform tomography or wave field inversion) falls into this second category. The particular propagation model used in that case is a full wave propagation model, *e.g.* the Helmholtz equation for acoustics in the frequency domain, Maxwell's equations for electromagnetics and the Navier equation for elastodynamics, at the opposite for example of ray tomography which are based on eikonal equations, *i.e.* on asymptotic approximations of the full model [10, 126, 206]. Considering a full model instead of a simplified one is computationally much more expensive but also promises a much higher resolution, as sub-wavelength accuracy is expected [7, 39, 54, 198, 122]. Another specificity of full waveform inversion is its brute force approach to the inverse problem: the mismatch is simply minimized w.r.t. the coefficients appearing in the full model. While simple in principle, this optimization nevertheless remains challenging: firstly because it is hard to design a meaningful mismatch functional for oscillatory signals such as wave scattering data [19, 124, 166, 203]; secondly because observed data are contaminated by noise, a perfect match with the simulated data is therefore often impossible and

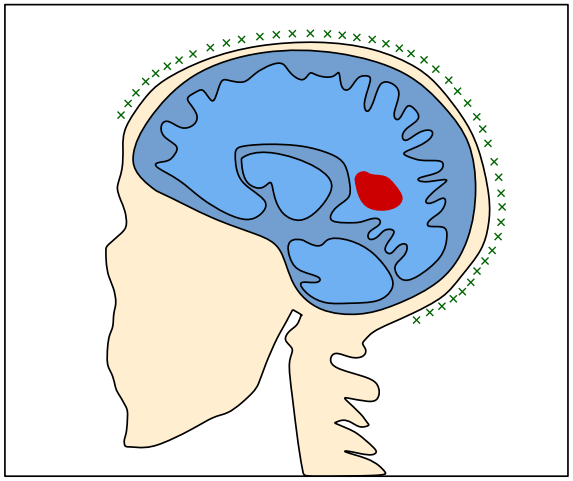
does not mean that the true coefficients are recovered anyway; finally because the coefficients are unknown at every location of the sample volume under investigation, hence large-scale optimization methods must be used. However, thanks to ever increasing available computational power, full waveform inversion is now becoming feasible and is therefore being investigated for a growing number of engineering applications. Three of them are described in Figure 1.



(a) Marine seismic survey. These surveys aim to image the structure and nature of the rock layers and the reservoirs beneath the seabed. Sound waves are sent into the earth from an excitation source (■), which is pulled in the water behind the survey vessel. Reflections from the different structures beneath the crust are traveling back to the surface and are recorded on a receiver streamer (•) towed behind the excitation source. Receivers can also be placed on the seafloor, using ocean-bottom cables or nodes.



(b) Ground-penetrating radar. This geophysical method uses radar pulses to image the subsurface media (soil, ice, fresh water, *etc.*) and its underground man-made utilities (concrete, pipes, cables, *etc.*). Similarly to seismic surveys, electromagnetic waves are generated either on-land by the cart transmitter (×), or inside a bore-hole (■) and the reflections from heterogeneities are recorded either at a transmitter (×) or in another parallel bore-hole (•). Air-launched antennas can also be used instead of an on-ground cart, but the signal strength is then weaker.



(c) Structural brain imaging. Neuro-imaging primarily focuses on identifying brain structures (lesions, vascular diseases, tumors, *etc.*) and is particularly relevant for rapid diagnosis and treatment of cerebrovascular accidents. The principle is again the same: a transmitter is used to generate an incident wave whose reflections are recorded at all the other transmitters (×). For this medical application, both electromagnetic and acoustic waves can be used.

Figure 1: Practical application examples for full waveform inversion.

Full waveform inversion originates from seismic exploration and in particular from the oil and gas industry for which the quantitative characterization of the subsurface geological layers is pivotal. First full waveform inversion attempts were performed in the time domain [178] and have been made possible thanks to the adaptation of adjoint state methods to geophysical problems [28]. Unfortunately when the starting model is not accurate enough, local optimization procedures may be trapped into local minima. Frequency domain approaches have then been developed, among others to tackle this issue [146, 148]. Frequency domain approaches enable to mitigate the non-linearity of the mismatch functional by inverting frequencies sequentially, from the lowest to the highest. Frequency domain methods can also benefit from the illumination redundancy provided by the acquisition configuration and consequently only consider a few frequencies, provided these frequencies are appropriately selected [171]. These numerous contributions allowed full waveform inversion to be successful for several real seismic applications [31, 132, 149, 157, 188]. The success of full waveform inversion for seismic exploration has contributed to its development for other application fields such as ground-penetrating radar or medical ultrasounds. Similarly to seismic exploration, ground-penetrating radar is a geophysical method to image the subsurface, but which uses electromagnetic excitation instead of acoustic excitation. Electrical properties (*e.g.* permittivity and conductivity) instead of mechanical properties (*e.g.* density and Lamé parameters) are therefore imaged through a ground-penetrating radar survey. Consequently, it is particularly efficient to detect soil water content or metallic inclusions at relatively shallow depth (typically few meters), while seismic exploration is meant to determine geological layers at larger scales (typically from tens to thousands of kilometers). In that context, full waveform inversion has first been applied synthetically to crosshole ground-penetrating radar data that aimed at locating water-filled tunnels [46]. A new bottleneck that appears when imaging electromagnetic properties is the large sensitivity discrepancy between permittivity and conductivity. This discrepancy forces the use of a differentiated treatment of the permittivity and the conductivity updates [46, 97, 112]. Nevertheless, full waveform inversion has also been successfully applied to observed ground-penetrating radar data from real field sites [47, 89, 90, 120]. As far as medical imaging is concerned, full waveform inversion has first been applied in two dimensions for breast tumors detection through ultrasound imaging [147]. The methodology was later expanded to three dimensions [108, 161] and to others medical applications such as muscle, tendon, ligament or cartilage imaging [13, 197] or the rapid diagnosis of brain stroke and head trauma [35, 68, 185]. A last application field which starts to investigate full waveform inversion is non-destructive testing, in particular for the assesment of masonry and concrete structures, with either acoustic or electromagnetic waves [6, 85, 92, 95, 139]. Over the past years, full waveform inversion has proven to be a powerful imaging method despite its huge computational cost. Nevertheless, many challenges remain, mostly to reduce this cost. Nowadays, full waveform inversion is still a dynamic field of investigations.

Mathematical problem statement

Full waveform inversion relies on a full wave propagation problem

$$F(m)u = f. \tag{1}$$

This propagation problem is formulated through a partial differential operator F, whose expression depends on some model parameters m which are precisely the sample properties to be determined. The excitation source f is here considered to be known, even though it might not always be the case. The solution to this problem is called a wave field u. It is for example a pressure field, an electric field or a displacement field. It must be a physically measurable quantity because it is compared to the observed data d_0. More specifically, the observed data are compared to a restriction/projection of the simulated wave field, because this wave field is only measured at a finite number of receivers in real acquisition setups. This restriction operator is denoted by R while the corresponding simulated data are denoted by d, and the exact model parameters m^* should then verify

$$d(m^*) := Ru(m^*) = d_0. \tag{2}$$

There is however no guarantee that the above equation has a unique solution, *i.e.* there might exist several distinct model parameters statisfying (2). Moreover in the presence of noise on the observed data, it is also possible that there is not even one solution. Absence of any exact solution can also be a consequence of wave propagation modeling assumptions. For these reasons, the above equation can be classified as an ill-posed problem [82]. Full waveform inversion therefore proposes instead to match the observed data d_0 and the simulated data d through the minimization of their distance

$$m^* = \arg\min \operatorname{dist}\left(d(m), d_0\right). \tag{3}$$

Full waveform inversion can then be expressed exclusively from the point of view of optimization as the minimization of a performance functional J (also called the misfit), defined as

$$J(m) := \operatorname{dist}\left(d(m), d_0\right). \tag{4}$$

In the aim of two or three dimensional imaging, the size of the model parameters space generally prevents the use of global optimization methods. Global optimization is only tractable in specific cases where a lot of *a priori* information is available, enabling the model parameterization to be very coarse [36, 37, 57, 160, 164]. In any other situation, large-scale optimization methods must be used. The latter are based on a local expansion of the misfit J, around the

current model estimate m, in some perturbed model direction δm

$$J(m+\delta m) \approx J(m) + \{D_m J\}(\delta m) + \frac{1}{2}\{D^2_{mm} J\}(\delta m, \delta m). \tag{5}$$

This expansion can be written, as above, in terms of the first and the second order directional derivatives $D_m J$ and $D^2_{mm} J$, but it can also be written in terms of the gradient j' and the Hessian operator H once an inner product $\langle \cdot, \cdot \rangle_M$ is chosen for the model space M [16, 76]

$$J(m+\delta m) \approx J(m) + \langle j', \delta m \rangle_M + \frac{1}{2} \langle H\delta m, \delta m \rangle_M . \tag{6}$$

The pure Newton direction p_N is then defined as the minimizer of this local quadratic expansion

$$p_N = \arg\min_{p \in M} \left[J(m) + \langle j', p \rangle_M + \frac{1}{2} \langle Hp, p \rangle_M \right] \tag{7}$$

which is also the solution of a linear system

$$Hp_N = -j'. \tag{8}$$

Local optimization techniques are iterative processes where the model parameters are updated based on this direction, or its approximations [130].

A typical full waveform inversion iteration loop therefore consists in several steps, illustrated in Figure 2. Starting with some initial guess for the model parameters m, the corresponding wave field u is computed through the wave propagation problem (1). The simulated data d are then mathematically extracted from that simulated wave field through the restriction

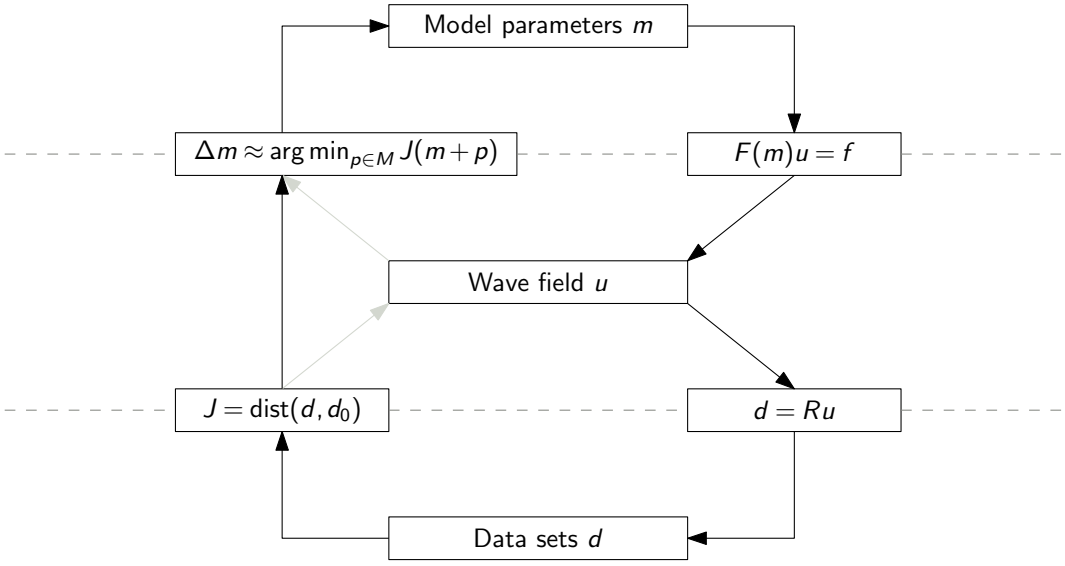


Figure 2: Schematic illustration of a typical full waveform inversion iteration loop. The derivatives of the misfit functional J are computed through the adjoint state method, hence the gray arrows.

operator R, mimicking a physical recording process. To compare them with the observed data d_0, a distance functional is then defined and the direction is computed as *approximately* minimizing this distance; while what is specifically meant by *approximately* depends on the chosen optimization method.
From this short introduction, it appears that three types of quantities are involved in a full waveform inversion workflow: data sets, wave fields and model parameters. In order to build an effective method, a particular attention must be given to each part. In a nutshell, data sets need to be compared in a consistent way, but defining what is a *close match* for oscillatory signals is not trivial; wave fields require the use of fast solvers, as their computation is the main bottleneck of the loop; and model parameters should be updated appropriately, in particular to reach the minimizer as soon as possible and to take into account any available *a priori* knowledge or physical constraints. More details about these three parts are given in the following paragraphs.

Data set space Wave scattering data are oscillatory by nature and consequently the misfit quantifying the discrepancy features local minima [19, 124, 166, 203]. Classically, this misfit is simply computed as the least squares norm of the residuals between measured and simulated data. Unfortunately, this simple distance leads to poor results if the starting model parameters are not accurate enough. Several distances or performance functionals have been tried in place of this least squares distance [30, 32, 191, 115, 168, 98, 201]. For example, the use of optimal transport yields significant progress, as it is a more suitable tool to compare oscillatory signals [115, 201]. Alternatively, performance functional that are not based on a specific distance in the data space have been proposed: for example using convolutional filters designed to transform the simulated data into the observed data [191]. In that case, the quantity to be minimized then reflects how far the matching filter is from the unit filter.

Wave field space During the local optimization process, full waveform inversion requires a large number of direct solutions to the full wave propagation model. An efficient inversion algorithm must therefore rely on an efficient forward solver. As far as wave scattering is concerned, solutions can be computed either in the time domain or in the frequency domain. Intuitive interpretation of the wave field is of course easier in the time domain. For example, first or late arrivals can be picked and processed separately. Computationally speaking, explicit time-stepping methods allow massive parallelism [64, 186, 202] but also require a lot of memory as the wave fields must be memorized for all the time steps. At the opposite, wave fields are lighter in the frequency domain because only a few frequencies are necessary. Moreover, scale separation can be performed by inverting low frequencies first [18, 22]. There is however also drawbacks to the frequency domain approach. Firstly, interpretation of the frequency domain trace is hard and picking specific events is impossible. Secondly, forward

solvers at high frequencies are challenging [48] and can not by easily parallelized [17, 81]. The frequency domain approach, related to the Fourier transform, can be generalized by using instead the Laplace transform. Through this substitution, the pulsation becomes complex-valued and its imaginary part introduces a constant dissipation, which mutes late arrivals. Laplace domain inversions are interesting because they allow to reconstruct time data traces progressively, from early to late arrivals, by progressively decreasing the dissipation. Moreover, for large dissipation, the forward modeling can be accurately calculated using a coarser grid than that of the Fourier domain [165, 167].

Model parameters space At the heart of full waveform inversion lies a local optimization algorithm. As mentioned previously, global optimization techniques should ideally be used but the typically very high dimension of the search space prohibits their use and only local optimization tools can practically be employed, with care. Local optimization methods for full waveform inversion typically involve the Hessian operator or one of its approximations because the inverse Hessian plays a crucial role in the reconstruction, in addition to offering the possibility to account for coupling effects between parameter classes for multi-parameter inversion [18, 133, 134, 148, 201]. State-of-the-art methods rely on the quasi-Newton l-BFGS algorithm, which implicitly builds an approximation of the inverse Hessian operator from l previously saved gradients and model parameters [130]. However, it has been illustrated that on some specific cases involving multiple reflections, such quasi-Newton methods fail to converge where Newton methods do succeed [118]. Hence the full-Newton method is now getting more and more attention. The building blocks of these local optimization techniques are the gradient and the Hessian operator, which can be computed with a minimal computational effort using the adjoint state method. A very common way to modify these two kernels is to modify the performance functional, as discussed previously, towards a nearly convex misfit without deep local minima. It is however not the only way: indeed both the gradient and the Hessian operator are defined with respect to an inner product. Hence changing appropriately this inner product will also modify advantageously the kernels [1, 2, 3, 210]. Another pivotal issue that can be handled through model space techniques is the mitigation of noise on the reconstruction image. Very frequently, prior information on the model parameters is available. This information can then for example be used to augment the misfit functional with a penalization term in the spirit of Tikhonov regularization [181]. The role of this penalization term is to promote model parameters that have some desired properties. To drive away from this prior guess, the observed wave scattering data should really oppose the simulated data. If the discrepancy is not strong enough, then the optimization algorithm will rather enforce the prior properties.

Dissertation goals

The global objective of this thesis is to contribute to the development of imaging methods through full waveform inversion. More precisely, the thesis focuses on the model parameter space aspect of the problematic. Interestingly, model space contributions are relevant regardless of whether the wave fields are computed in the time or the frequency domain. Model parameters however depend on the underlying physics: acoustics, elastodynamics or electromagnetics. In this thesis, propagation problems are solved in the frequency domain, using the most widely used propagation model for each physic, *i.e.* the Helmholtz equation, the Navier equation and Maxwell's equations respectively, and the associated usual medium parameters. In order to demonstrate the efficiency of the proposed inversion strategies, three guiding numerical case studies will be investigated in detail throughout this whole manuscript and will thus provide a natural link between all the chapters. These three reference case studies are introduced here below. All are two dimensional and synthetic, meaning that the data is created by a numerical model rather than being acquired experimentally. In addition, all inversion procedures are based on the simplest wave propagation model in the frequency domain, namely the Helmholtz equation.

Case study 1: Marmousi model

The first case study is based on the Marmousi model [187]: a synthetic model, created by the Institut du Prétrol Francais in 1988, which represents the velocity of acoustic waves inside the geological layers composing the earth below the seafloor. It was designed to be geologically plausible, and it is actually based on a profile through the north Quenguela trough in the Cuanza basin in Angola. The Marmousi velocity distribution, shown in Figure 3, is complex: it contains many reflectors, steep dips, faults and strong velocity gradients in both vertical and lateral directions. Moreover, a water layer, which can reasonably be considered to be known, is added above the seabed. Overall, the model is approximately 9 [km] wide and 3 [km] deep. Reconstructing an accurate earth image therefore requires advanced processing techniques, such as full waveform inversion. In particular, anticline areas and salt domes are of great interest as they might hide hydrocarbon reservoirs underneath, because they are particularly efficient traps. During a marine seismic survey, sound waves are generated inside the water layer thanks to an excitation source, which is pulled behind the survey vessel. Reflections from the seabed and the geological layers beneath are then recorded with a receiver streamer towed behind the excitation source, or with ocean-bottom cables or nodes. With such an acquisition, only surface data are obtained, which further complicates the inversion process, because the illumination is incomplete, *i.e.* the earth layers to be imaged are viewed under a limited number of angles. In addition, the low frequency response is often missing, because it can not be measured in practice, due to the *ghost reflection*, *i.e.* the direct reflection of the

excitation pulse on the water free surface [174]. Since its inception in 1988, the Marmousi model has become an industry and academy standard for acoustic seismic imaging, which makes it particularly relevant for a first case study.

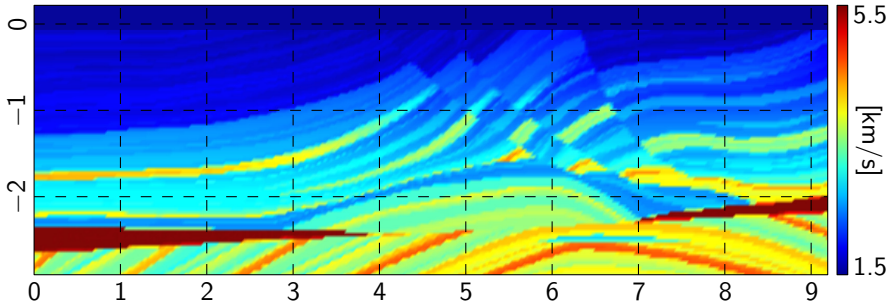


Figure 3: The Marmousi velocity model. Two anticline areas are located around $(x, z) = (5.5, -2.0)$ [km] and $(x, z) = (6.5, -2.5)$ [km], while two triangular salt structures lie on both sides, below $z = -2$ [km]. In addition, the model also contains many reflectors, steep dips and faults [110]. Distances are expressed in kilometers.

Case study 2: T-shaped reflectors

The second case study has been first introduced by [118]. It consists in a near-surface land survey, whose goal is to image two T-shaped underground concrete structures, as shown in Figure 4, in the spirit of non-destructive testing. The investigation depth is 3 [m] while the width is 30 [m]: the aspect ratio and the imaged area are thus very different from the Marmousi model. This second case study does not aim to be realistic though: it is rather used firstly because the proximity of both reflectors generates lots of multiple reflections in between, which are typically hard to interpret; and secondly because the high-velocity values of the two concrete structures are difficult to reach due to their strong contrast with the background. For this case study, the acquisition is three-sided: excitation sources and receivers are located on the surface and inside two boreholes on both sides of the survey area. Moreover, a horizontal layer reflector lies below the structures. This light reflector generates sound waves traveling from the bottom back to the surface, hence illumination is nearly complete for this case study. The only two remaining challenges are multiple reflections and strong velocity contrast: the two specific motivations for the use of this test case.

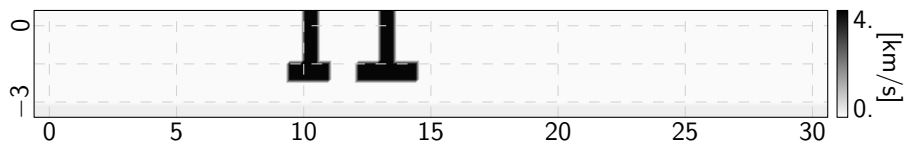


Figure 4: The T-shaped concrete reflectors velocity model. The two concrete structures exhibit a strong contrast with the background. Their closeness also generates strong multiple reflections in between [118]. Distances are expressed in meters.

Case study 3: Dissipative crosses

The third case study is an academic example which is frequently used to compare different multi-parameter inversion schemes, as a first step to use full waveform inversion for ground penetrating radar applications [97, 111]. In that context, electromagnetic fields are generated in the soil to image its structure through its electrical properties, typically the permittivity and the electric conductivity. The permittivity is related to the electromagnetic wave velocity, while the conductivity introduces energy dissipation. Soils and man-made structures under investigations typically exhibit large dissipation at the operating frequencies of ground-penetrating radar, such that neglecting the conductivity is not a successful strategy. Note however that this is not systematically the case: for seismic inversion for example and thus for the first two case studies, attenuation can be safely discarded as a first approximation. Basically, the main difficultly of any multi-parameter inversion is *cross-talk*, in which one parameter is mistakenly updated to account for a discrepancy between observed and simulated data caused by another parameter. Double-parameter full waveform inversion is the academy standard for ground-penetrating radar, such that it makes sense to consider an example from this community to illustrate the challenges related to multi-parameter inversions. The electrical property distribution, given in Figure 5, is relatively simple: two crosses with permittivity and conductivity anomalies, embedded in an homogeneous dissipative background. Strong property contrasts are again enforced, for consistency with real electrical targets. Since the purposes of this case study is multi-parameter inversion, any further complexity is discarded. In particular, a perfect circular acquisition system is considered to provide a full illumination of the scene and thus discard any shadowing issues.

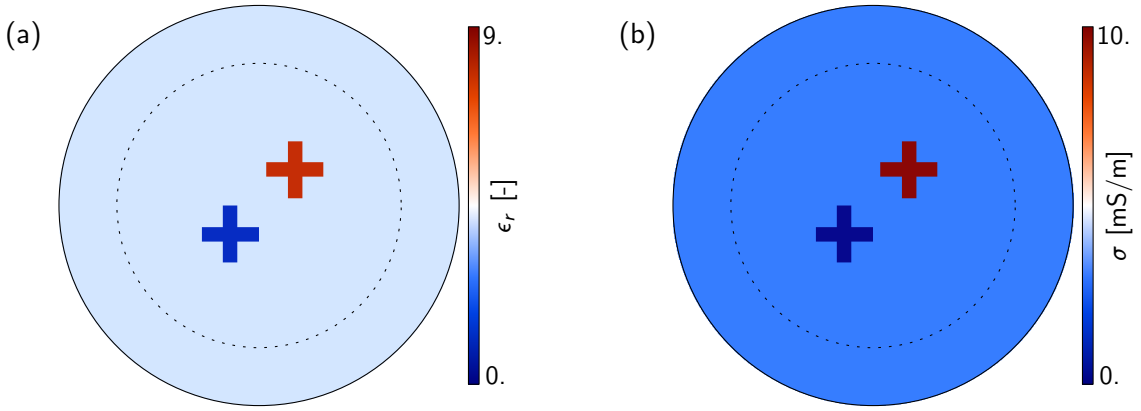


Figure 5: The dissipative crosses electrical property distribution: relative permittivity (a) and electric conductivity (b). Crosses exhibit strong electrical properties contrasts. Inversion aims to reconstruct both parameter distributions. Outer circle (solid line) radius is 7 [m]. Acquisition circle (dotted line) radius is 5 [m]. Crosses equivalent radius is 1 [m].

Dissertation outline

This dissertation is divided into three chapters, each of them mixing original contributions and reminders of state-of-the-art techniques when it is necessary to preserve the continuity of the presentation. Chapters are also organized such that the different concepts are ordered as they appear in the full waveform inversion loop. Their content are summarized here below.
Chapter 1 is dedicated to the efficient computation of sensitivities, *i.e.* first and second order derivatives and the corresponding gradient kernel and Hessian operator, thanks to a procedure called the *adjoint state method*. It is derived, implemented and validated for different types of model parameters and for multiple wave propagation models. The same formalism is used for all wave physics such that the equivalence between them can easily be drawn. The effect of inner product modifications on the derivatives is studied in this chapter as well.
Chapter 2 then naturally focuses on using these sensitivities to build local optimization algorithms. More specifically, a line search and several trust-region variants of the steepest descent, the limited memory BFGS algorithm and the inexact Newton method are presented and compared in the context of the three reference case studies. A strong emphasis is given to the inner product choice. For example, its link with preconditioning the update direction and its implication in the trust-region constraint are highlighted.
Chapter 3 finally studies the influence of noisy data on the reconstruction. In particular, the *per se* regularization properties of inner product preconditioned optimization methods are investigated and compared to conventional regularization strategies. Moreover, a more sophisticated adaptive inner product is introduced. It takes inspiration from the nonlinear anisotropic diffusion filters developed in the field of image processing and acts as a structure-preserving smoothing operator.
As key information on the performance of the proposed local optimization strategies, the accuracy, the robustness to noise, and the convergence speed of each numerical implementation is carefully analyzed throughout the work.

Original contributions

Hereafter is a list of contributions that are considered to be original:

- Chapter 1:
  (a) The derivation of the adjoint state method for general boundary value problems using a continuous formalism in order to highlight the role of the boundary terms.
  (b) The systematic application of the adjoint state method to acoustic, electromagnetic and elastodynamic wave scattering problems with impedance boundary conditions, making use of the similarities between the three problems.

- Chapter 2:
  (a) The extensive comparison of line search and trust-region variants of the steepest descent, the limited memory BFGS algorithm and the inexact Newton method in the context of full waveform inversion.
  (b) The application of inner product preconditioining to the limited memory BFGS algorithm and the inexact Newton method.
  (c) The use of inner product preconditioining to balance sensitivity disparities in multi-parameter class inversions.
- Chapter 3:
  (a) The study of regularization properties of inner product preconditioned optimization methods.
  (b) The introduction of a structure preserving smoothing inner product.

In addition, an open-source C++ code, called `GmshFWI`, has been developed to implement all the described algorithms. Its structure is inherited from the mathematical statement of the problem as it is divided into the same three parts: the model parameter space with its inner product; the wave field space with its forward problem; the data space with its distance functional. A particular emphasis was given to flexibility, through the definition of abstract interfaces, to allow for a maximal number of investigations, for this thesis and beyond. This code is also strongly based on `GmshFEM`: an efficient C++ finite element library [158] based on the application programming interface of `Gmsh` [61]. Implementation details and full numerical case studies ready for testing are available online (`https://gitlab.onelab.info/gmsh/fwi`).

The main original contributions of this thesis have been presented in the following peer-reviewed journals and conference proceedings:

- **Journal papers:**
  - X. Adriaens, F. Henrotte, and C. Geuzaine. "Adjoint state method for time-harmonic scattering problems with boundary perturbations". In: *J. Comput. Phys.* 428 (Mar. 2021), p. 109981
  - X. Adriaens, L. Métivier, and C. Geuzaine. "Inner product preconditioned trust-region methods for frequency-domain full waveform inversion". In: *Review* ()

- **Conference proceeding:**

  - X. Adriaens, L. Métivier, and C. Geuzaine. "A trust-region Newton method for frequency-domain full-waveform inversion". In: *First Int. Meet. Appl. Geosci. Energy Expand. Abstr.* Vol. 2021-Septe. Society of Exploration Geophysicists, Sept. 2021, pp. 757–761

- **Conference presentations:**

  - X. Adriaens, L. Métivier and C. Geuzaine, C. 2021. "A trust-region Newton method for frequency-domain full waveform inversion". At: *1st International Meeting for Applied Geoscience and Energy (IMAGE)*. Denver, United States. (remote)

  - X. Adriaens and C. Geuzaine. 2021. "Simultaneous permittivity and conductivity full waveform inversion in the frequency domain". At: *16th international workshop on Optimization and Inverse Problems in Electromagnetism (OIPE)*, Szczecin, Poland. (remote)

  - X. Adriaens and C. Geuzaine. 2021. "Simultaneous permittivity and conductivity full waveform inversion in the frequency domain". At: *12th international symposium on Electric and Magnetic Fields (EMF)*, Aix-en-Provence, France. (remote)

  - X. Adriaens, L. Métivier and C. Geuzaine. 2022. "Inner product preconditioned trust-region methods for frequency-domain full waveform inversion". At: *8th international conference on Advanced COmputational Methods in ENgineering (ACOMEN)"*. Liège, Belgium.

Chapter 1

Adjoint state method

Knowing how the solution to time-harmonic wave scattering problems depends on medium properties and boundary conditions is pivotal for full waveform inversion and more generally for any wave-based inverse problem. This first chapter is devoted to the exposition of a computationally efficient method, called the adjoint state method, that allows to quantify the influence of media properties, directly and through boundary conditions, in the study of acoustic, electromagnetic and elastodynamic time-harmonic waves. More specifically, this method is an efficient procedure to compute derivatives, gradients and Hessian operators. Firstly, the adjoint state method is derived for general boundary value problems. A continuous (rather than discrete) formalism is adopted in order to highlight the role of the boundary terms. Then, the method is applied systematically to acoustic, electromagnetic and elastodynamic scattering problems with impedance boundary conditions, making use of the similarities between the three problems. Finally, numerical examples solved using the finite element method are presented to demonstrate the validity of the proposed method.

Highlights

- Definition of directional derivatives, gradient kernels and Hessian kernels
- Derivation of the adjoint state method with boundary perturbations
- Application to time-harmonic scattering of acoustic, electromagnetic and elastic waves
- Numerical validation through an illustrative example

1.1 Introduction

The analysis of the influence of medium properties on wave fields is useful in many physical and engineering problems involving time-harmonic wave propagation. Among these problems

lie of course full waveform inversion for geophysical or medical imaging by acoustic [119, 142], electromagnetic [42, 145, 153] or elastodynamic [18, 148] waves but also invisibility cloaking [23, 93, 104] or the optimal design of acoustic liners [150], optical devices [74, 80, 96, 129, 169], vibrating structures [71, 182, 184], antennas [60, 62] and electromagnetic cavities [5], for example.

Most of these applications are naturally set in unbounded domains and involve complex geometries and multiple and/or inhomogeneous materials. Specific boundary conditions can however be used to reduce the computational complexity and make the numerical study tractable, by substituting an equivalent impedance boundary condition to volume scatterers with complex (or even unknown) properties [5, 23, 74, 93, 104, 119, 150, 169], or by truncating unbounded domains with transparent absorbing boundary condition [109, 163, 170]. Boundary conditions contain in this case information about all regions left outside the model, and their influence is therefore crucial.

For optimization purposes, it is often asked to maximize/minimize some objective function of measurable quantities like pressure, electromagnetic or displacement fields. In imaging applications, this objective is the distance between predicted and experimental field measurements, whereas in design applications, the objective for example includes resonant frequencies and eigenmodes of vibrating structures, antennas or cavities and in particular their discrepancy w.r.t. to some ideal values. Now, field propagation over the system can only be controlled through physical model parameters like the speed of waves or the geometrical shape of some regions, and the expression of the relationship between these parameters and the measurable field requires solving partial differential equations, which is computationally expensive. As mentioned in the general introduction, the very high dimension of the search space in such problems prohibits using global optimization techniques, and makes gradient-based and local optimization algorithms very attractive. Such algorithms require however to compute sensitivities efficiently, with adjoint methods for instance.

There are two ways of introducing the adjoint state method for problems involving partial differential equations. The first approach consists in discretizing the problem first, and to compute sensitivities then using the tools of linear algebra. Advantages of this approach are the simplicity of the formulation and an easy numerical validation (because of the finite dimension of the problem). With this approach, often referred to as *first discretize, then optimize* [76], it is however hard to track the individual influence of the different parameters, or to distinguish the effect of bulk and boundary terms. The second approach consists in computing sensitivities analytically at the continuous level before discretizing the problem. This formalism often referred to as *first optimize, then discretize* [76], is better suited for an intuitive interpretation of the equations and allows dealing with boundary and bulk contributions separately.

The algebraic adjoint state method (*first discretize, then optimize*) with boundary terms has been used in several shape design applications [5, 60, 62, 96]. In this paper, however, the

first optimize, then discretize approach has been preferred, and an application-independent formalism well-suited for topological variation in time-harmonic wave scattering problems has been developed. A similar application-independent framework with partial differential equations constraints has been presented for optimization applications in [76], which did not however pay any particular attention to the boundary aspects. Extensions to include boundary variations have been proposed by [63, 74] and applied to shape sensitivities in aerodynamics and optics, respectively. The formalism proposed in this chapter is based on [63, 74], but it is applied to time-harmonic wave scattering problems in view of practical engineering computations and in particular of full waveform inversion. In the context of such time-harmonic scattering problems, sensitivities to bulk topological parameters have been analyzed extensively in [18, 42, 54, 119, 145, 142, 148, 196, 210], but sensitivities to boundary topological parameters have not been investigated yet to the best of my knowledge. Obtaining gradients from sensitivities is straightforward with the algebraic adjoint state method, but can be rather more intricate with the *first optimize, then discretize* approach, especially when dealing with non-standard gradients like Sobolev gradients [127, 154] that appear when considering non-standard inner product for the model parameter space. Such developments have been reported in [42, 127, 154, 155, 210] for bulk sensitivities, and they are extended in this chapter to include boundary sensitivities. The use of non-standard gradients in time-harmonic scattering problems has proven efficient in recent publications [196, 210]. It is applied in this framework to wave propagation problems from three distinct physics (acoustics, electromagnetics and elastodynamics) in order to highlight the vast similarities between them. This chapter focuses on topological model parameters rather than shape model parameters, which have been studied in [77] from the perspective of differential forms.

The chapter is organized as follows. In Section 1.2, the mathematical framework is made explicit and the quantities to compute are discussed. The direct method to compute these quantities is given in Section 1.3, then the adjoint state method is derived for Gâteaux derivatives without and with boundary contributions in Subsection 1.4.1 and 1.4.2 respectively, and for gradient kernels and Hessian kernels in Subsection 1.4.3 and 1.4.4 respectively. Then in Section 1.5, the results are applied to time-harmonic acoustic, electromagnetic and elastodynamic wave propagation problems with impedance boundary conditions. A formal link is established between the three problems and the choices required by the adjoint state method are discussed simultaneously. In Section 1.6, the method is applied to a complete two dimensional sensitivity problem, typical of full waveform inversion problems. The correspondence between the adjoint state method and a naive approach, based on a brute force approximation of the derivatives by finite differences, is verified for volume and surface contributions and the results are discussed. Finally, in Section 1.7, the three reference case studies are briefly considered, in preparation for the following chapter.

1.2 Problem statement

Consider a physical system for which a *state space model* is known, usually under the form of a set of partial differential equations. The input m to this model is called *model parameter* and is a vector belonging to the *model space* M. The output u of the model is called *state variable* and belongs to a function space $U(\Omega)$ called the *state space*. The latter is assumed to be a sufficiently regular subset of the space of square integrable function $U_2(\Omega)$ defined on an open bounded set $\Omega \subset \mathbb{R}^n$, because the derivatives of the state variable u will be needed. In the sequel, all spaces of square integrable functions will be denoted with a subscript '2'.
The state base model

$$\begin{cases} F(u,m) &= f, \\ B(u,m) &= g \end{cases} \tag{1.1}$$

is the physical link that implicitly defines the state $u(m)$ as a function of the input m. This system is assumed to be well-posed. The first equation involves a *direct state operator* $F : U(\Omega) \times M \to U_2^\dagger(\Omega)$ whose co-domain is a function space of square integrable functions $U_2^\dagger(\Omega)$. This space $U^\dagger(\Omega)$ is the analog of $U(\Omega)$ for the *adjoint state variable* (to be defined later), hence the notation. The second equation in (1.1) involves a *direct state boundary operator* $B : U(\Omega) \times M \to B_2(\partial\Omega)$ whose co-domain $B_2(\partial\Omega)$ is again a function space of square integrable functions, and where $\partial\Omega$ stands for the boundary of the domain Ω. Such an operator involves some traces of the wave field, because it acts from a bulk function space to a boundary function space. Hence, it will rather be called a *boundary trace-based operator*; in opposition to a *purely boundary operator* which acts from a boundary function space to another boundary function space. For the sake of conciseness, the model dependency $u(m)$ will not always be indicated explicitly. A state noted u will thus always be assumed to be a solution of the system (1.1).
Besides (1.1), the second element of the theoretical setting is a real-valued functional of the state variable $u(m)$

$$J(m) := H(u(m)) + K(C(u(m), m)) \tag{1.2}$$

called the *performance functional*. As for (1.1), the performance functional J is decomposed into two terms: a *bulk performance functional* $H : U_2(\Omega) \to \mathbb{R}$ depending on the value of the state variable $u(m)$ in the bulk of the domain Ω, and a *boundary performance functional* $K : C_2(\partial\Omega) \to \mathbb{R}$ depending on a specific boundary trace-based operator applied on $u(m)$. The boundary trace-based operator $C : U(\Omega) \times M \to C_2(\partial\Omega)$ is called *performance boundary trace-based operator and, similarly to B, its co-domain $C_2(\partial\Omega)$ is a space of square integrable functions. Because this performance trace-based operator* might also involve spatial derivatives, it is possible that the state and model spaces must be chosen to be more regular than what is required by the state base model (1.1) alone.
Quantifying analytically and numerically the variation of the performance functional J under

a perturbation of the model parameter m is the subject of this chapter. This variation can be expressed by two different derivatives. Firstly, given a perturbation δm of the model parameter, the *directional derivative* or *Gâteaux derivative*

$$\{D_m J(m)\}(\delta m) := \lim_{\epsilon \to 0} \frac{J(m + \epsilon\, \delta m) - J(m)}{\epsilon} \tag{1.3}$$

gives the variation of the performance functional J for an arbitrarily small modification of the model parameter m in the direction δm [76]. A *partial directional/Gâteaux derivative* is defined similarly for functionals with more than one argument. For instance, the performance functional (1.2) could also be regarded as a functional with two arguments $J(u, m)$ and the partial Gâteaux derivative with respect to the argument m is then denoted by $\{\partial_m J(u, m)\}(\delta m)$ with the usual ∂ symbol.

Considering that the full-fledged notations for Gâteaux total and partial derivatives are rather heavy, shorthands shall be used in the sequel wherever no confusion is possible. The *total* derivative of J will be noted $\delta J := \{D_m J(m)\}(\delta m)$ with the symbol δ to recall that it represents the Gâteaux derivative of J associated to the parameter perturbation δm. The mute arguments of the partial Gâteaux derivatives, on the other hand, might be omitted after their first introduction, *e.g.* , $\{\partial_m J\}(\delta m) := \{\partial_m J(u, m)\}(\delta m)$.

Most of the time however, a privileged directions δm does not make sense for the definition of sensitivity. In that case, the concept of gradient kernel must be preferred. A gradient kernel is an element of the model space M of which each component quantifies the variation of J for an arbitrarily small perturbation of m along that particular axis only. Mathematically, these *Fréchet-Wirtinger's gradient kernels*, or simply *gradient kernels*, are denoted by j' and defined by [16, 76]

$$\mathrm{Re}\,\langle j'(m), \delta m\rangle_M := \{D_m J(m)\}(\delta m), \quad \forall \delta m \in M \tag{1.4}$$

where $\langle \cdot, \cdot \rangle_M$ is an inner product on M. The real part must be taken as the model parameter m (and the inner product) might be complex-valued while the performance functional is real-valued. For the sake of compactness, the symbol 'Re' is omitted in this thesis. By definition, gradients kernels depend thus on the choice of a specific inner product in the function space under consideration. Conceptually, the gradient kernel is a vector pointing in the direction of steepest ascent, *i.e.* the direction that produces the largest increase of the performance functional among all directions of fixed arbitrarily small lengths. This direction is particularly relevant in optimization processes, and it can be interpreted as follows in the case of the inner product of square integrable functions. Consider that the model parameter m is perturbed at a single point $\boldsymbol{y}$, *i.e.* $\delta m = \delta(\boldsymbol{x} - \boldsymbol{y})$ with $\delta(\boldsymbol{x})$ denoting the Dirac delta function. The derivative δJ is then given by $\delta J = j'(\boldsymbol{y})$, which is indeed the evaluation at point $\boldsymbol{y}$ of the gradient kernel j'. This is why these kernels can be viewed as sensitivities.

When a more subtle description of the performance functional variation is required, second

order derivatives are often computed. The *second order directional derivative* is defined as the derivative of the first order directional derivative

$$\left\{D^2_{mm}J(m)\right\}(\delta m_1, \delta m_2) := \left\{D_m\left\{D_m J(m)\right\}(\delta m_1)\right\}(\delta m_2) \tag{1.5}$$

while the Hessian operator, denoted by $H(m)$, is defined by

$$\mathrm{Re}\left\langle H(m, \delta m_2), \delta m_1\right\rangle_M := \left\{D^2_{mm}J(m)\right\}(\delta m_1, \delta m_2), \quad \forall \delta m_1,\ \delta m_2 \in M. \tag{1.6}$$

Interestingly, the Hessian operator can also be expressed as the derivative of the gradient kernel. Indeed substituting (1.4) into (1.5) gives

$$\begin{aligned}\left\{D^2_{mm}J(m)\right\}(\delta m_1, \delta m_2) &= \left\{D_m \mathrm{Re}\left\langle j'(m), \delta m_1\right\rangle_M\right\}(\delta m_2) &(1.7)\\ &= \mathrm{Re}\left\langle\left\{D_m j'(m)\right\}(\delta m_2), \delta m_1\right\rangle_M &(1.8)\\ &:= \mathrm{Re}\left\langle \delta_2 j'(m), \delta m_1\right\rangle_M. &(1.9)\end{aligned}$$

Then relating (1.6) and (1.9) gives

$$\left\langle H(m, \delta m_2), \delta m_1\right\rangle_M := \left\{D^2_{mm}J(m)\right\}(\delta m_1, \delta m_2) = \left\langle \delta_2 j'(m), \delta m_1\right\rangle_M, \quad \forall \delta m_1 \in M \tag{1.10}$$

and consequently

$$H(m, \delta m_2) = \delta_2 j'(m). \tag{1.11}$$

For complex-valued model parameters, the Hessian operator is composed of both a linear part and an anti-linear part w.r.t. the model parameter perturbation δm_2. Solving the Newton system (8) with such a Hessian operator is impractical because linear system solvers are typically used. The best option in that case is then to consider the real and imaginary parts separately, or to neglect the anti-linear part of the Hessian [94, 121, 137]. For real-valued model parameters however, the Hessian operator is linear w.r.t. the model parameter perturbation δm_2. It can then be written through a Hessian kernel h' as $H(m, \delta m_2) := \left\langle h', \delta m_2\right\rangle_M$. An interpretation of this Hessian kernel can then be drawn from the above equation. Considering again a model parameter perturbation at a single point, *i.e.* $\delta m_2 = \delta(\boldsymbol{x} - \boldsymbol{y}_2)$, the variation $\delta_2 j'$ of the sensitivity is given by $\delta_2 j' = h'(\cdot, \boldsymbol{y}_2)$, which means that $h'(\boldsymbol{y_1}, \boldsymbol{y}_2)$ is the variation of the sensitivity at point $\boldsymbol{y}_1$ for a perturbation located at $\boldsymbol{y}_2$. The Hessian kernel is the sensitivity of the sensitivity and hence it is called a second order sensitivity.

1.3 Direct approach

The performance functional $J(u(m))$ depends on the model parameter m through the state u. Consequently, to obtain the Gâteaux derivative δJ in the direction δm, it is necessary to know the state derivative δu in that direction. This quantity is implicitly defined by taking the total derivative of (1.1) with respect to m. One has

$$\begin{cases} \{D_m F(u,m)\}(\delta m) &= 0, \\ \{D_m B(u,m)\}(\delta m) &= 0, \end{cases} \tag{1.12}$$

which writes, in terms of partial derivatives,

$$\begin{cases} \{\partial_u F(u,m)\}(\delta u) &= -\{\partial_m F(u,m)\}(\delta m), \\ \{\partial_u B(u,m)\}(\delta u) &= -\{\partial_m B(u,m)\}(\delta m). \end{cases} \tag{1.13}$$

This is a boundary value problem that can be solved for δu.
Using now (1.2), the total derivative δJ of the performance functional reads

$$\delta J := \{D_m J(m)\}(\delta m) \tag{1.14}$$

$$= \{D_u H(u)\}(\delta u) + \{D_C K(C(u,m)\}(\delta C) \tag{1.15}$$

and using the bulk and boundary performance gradient kernels $h'(u)$ and $k'(C(u,m))$, it can be rewritten as

$$\delta J = \langle h', \delta u \rangle_{U_2(\Omega)} + \langle k', \delta C \rangle_{C_2(\partial\Omega)}. \tag{1.16}$$

The second term can also be differentiated, so that the Gâteaux derivative δJ is explicitly expressed as a function of the parameter perturbation δm and the solution δu of the boundary value problem (1.13)

$$\delta J = \langle h', \delta u \rangle_{U_2(\Omega)} + \langle k', \{\partial_u C\}(\delta u) \rangle_{C_2(\partial\Omega)} + \langle k', \{\partial_m C\}(\delta m) \rangle_{C_2(\partial\Omega)}. \tag{1.17}$$

This way of computing the Gâteaux derivative δJ associated with the direction δm is called the *direct approach*.

1.4 Adjoint approach

The direct approach introduced in the previous section is rather straightforward, but it can reveal rather inefficient for some classes of problems, in which cases the adjoint approach is a powerful alternative. This approach is first presented without boundary perturbation, in order

to establish the fundamental concept of adjoint operators, and then generalized to problems with boundary perturbation.

1.4.1 Without boundary perturbation

The adjoint method is based on the adjoint operator of $\partial_u F(u,m)$ in (1.13), which is defined as the operator $\partial_u^\dagger F(u,m)$ fulfilling the integration by parts relationship

$$\left\langle u^\dagger, \{\partial_u F(u,m)\}(\delta u)\right\rangle_{U_2^\dagger(\Omega)} = \left\langle \left\{\partial_u^\dagger F(u,m)\right\}(u^\dagger), \delta u\right\rangle_{U_2(\Omega)} + \left[u^\dagger, \delta u\right]_{\partial_u F} \tag{1.18}$$

where $u^\dagger$ is called the adjoint state variable, and where the boundary term $\left[u^\dagger, \delta u\right]_{\{\partial_u F\}}$ is a differential expression involving δu and $u^\dagger$, integrated over the boundary $\partial\Omega$ and that depends only on the operator $\partial_u F(u,m)$, here abbreviated $\partial_u F$.
Disregarding boundary terms provisionally, the direct problem defined above reads

$$\begin{cases} \{\partial_u F(u,m)\}(\delta u) = -\{\partial_m F\}(\delta m), \\ \delta J = \langle h', \delta u\rangle_{U_2(\Omega)} \end{cases} \tag{1.19}$$

and it is now shown that it admits an equivalent adjoint problem

$$\begin{cases} \left\{\partial_u^\dagger F(u,m)\right\}(u^\dagger) = h', \\ \delta J = -\left\langle u^\dagger, \{\partial_m F\}(\delta m)\right\rangle_{U_2^\dagger(\Omega)}. \end{cases} \tag{1.20}$$

The equivalence is obvious whenever the boundary term $\left[u^\dagger, u\right]_{\partial_u F}$ vanishes, as one has then, using (1.18),

$$\begin{aligned} \delta J = \langle h', \delta u\rangle_{U_2(\Omega)} &= \left\langle \left\{\partial_u^\dagger F\right\}(u^\dagger), \delta u\right\rangle_{U_2(\Omega)} \\ &= \left\langle u^\dagger, \{\partial_u F\}(\delta u)\right\rangle_{U_2^\dagger(\Omega)} = -\left\langle u^\dagger, \{\partial_m F\}(\delta m)\right\rangle_{U_2^\dagger(\Omega)}. \end{aligned} \tag{1.21}$$

It is here worth noting that the right-hand side of the direct problem appears in the evaluation of δJ in the adjoint problem, whereas conversely the right-hand side of the adjoint problem appears in the evaluation of δJ in the direct problem.
Deciding between the direct or the adjoint approach depends now on the respective numbers of performance functionals and model perturbations. In order to complete the evaluation of δJ, the direct problem (1.19) needs be solved once for each direction δm. The adjoint problem (1.21), on the other hand, needs be solved once for each value of the gradient kernels h', *i.e.* once for each performance functional J. As the direct and the adjoint problems imply solving linear systems of comparable size and complexity, the adjoint approach is preferred whenever there are more search directions than performance functionals to evaluate, and the

direct approach is preferred otherwise. In the context of full waveform inversion, the misfit between simulated and measured data is the only performance functional to be minimized and the adjoint method is thus systematically used.

1.4.2 With boundary perturbation

One now turns to the case of a problem with boundary perturbation. The vanishing of the boundary term that was assumed in the previous section is in practice a too stringent condition. A less restrictive and more general condition of existence for the adjoint problem consists in assuming there exist two trace-based operators

$$\left\{\partial_u^\dagger B(u,m)\right\}(\cdot) : U^\dagger \to C_2(\partial\Omega) \quad \text{and} \quad \left\{\partial_u^\dagger C(u,m)\right\}(\cdot) : U^\dagger \to B_2(\partial\Omega) \tag{1.22}$$

such that the boundary term in (1.18) verifies the identity

$$\begin{aligned}\left[u^\dagger, \delta u\right]_{\partial_u F} = &\left\langle \left\{\partial_u^\dagger B(u,m)\right\}(u^\dagger), \left\{\partial_u C(u,m)\right\}(\delta u)\right\rangle_{C_2(\partial\Omega)} \\ &- \left\langle \left\{\partial_u^\dagger C(u,m)\right\}(u^\dagger), \left\{\partial_u B(u,m)\right\}(\delta u)\right\rangle_{B_2(\partial\Omega)}.\end{aligned} \tag{1.23}$$

This is a fairly mild assumption of which the implied mathematical restrictions (*i.e.* the restrictions applying on the operators F, B and C) are discussed in detail in [63].
The *adjoint state variable* $u^\dagger$ is now the solution of the *adjoint problem with boundary perturbation*

$$\begin{cases} \left\{\partial_u^\dagger F(u,m)\right\}(u^\dagger) = h', \\ \left\{\partial_u^\dagger B(u,m)\right\}(u^\dagger) = k' \end{cases} \tag{1.24}$$

and the Gâteaux derivative of the performance functional (1.17) needs be reexpressed in terms of the adjoint state variable $u^\dagger$ and the adjoint operators as follows (1.22)

$$\begin{aligned}\delta J = \quad & \langle h', \delta u\rangle_{U_2(\Omega)} + \langle k', \{\partial_u C\}(\delta u)\rangle_{C_2(\partial\Omega)} \\ & + \langle k', \{\partial_m C\}(\delta m)\rangle_{C_2(\partial\Omega)} \end{aligned} \tag{1.25}$$

$$\begin{aligned}= \quad & \left\langle \left\{\partial_u^\dagger F\right\}(u^\dagger), \delta u\right\rangle_{U_2(\Omega)} + \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_u C\right\}(\delta u)\right\rangle_{C_2(\partial\Omega)} \\ & + \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m C\right\}(\delta m)\right\rangle_{C_2(\partial\Omega)} \end{aligned} \tag{1.26}$$

$$\begin{aligned}= \quad & \left\langle u^\dagger, \left\{\partial_u F\right\}(\delta u)\right\rangle_{U_2^\dagger(\Omega)} + \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \left\{\partial_u B\right\}(\delta u)\right\rangle_{B_2(\partial\Omega)} \\ & + \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m C\right\}(\delta m)\right\rangle_{C_2(\partial\Omega)} \end{aligned} \tag{1.27}$$

$$\begin{aligned}= - & \left\langle u^\dagger, \left\{\partial_m F\right\}(\delta m)\right\rangle_{U_2^\dagger(\Omega)} - \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \left\{\partial_m B\right\}(\delta m)\right\rangle_{B_2(\partial\Omega)} \\ & + \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m C\right\}(\delta m)\right\rangle_{C_2(\partial\Omega)} \end{aligned} \tag{1.28}$$

using successively (1.24), (1.23) and (1.13). This way of computing the Gâteaux derivative δJ associated with the direction δm is called the *adjoint approach*.

1.4.3 Gradient kernels

On basis of (1.28), one can show that the adjoint state $u^\dagger$ and the adjoint trace-based operators $\left\{\partial_u^\dagger C\right\}(u^\dagger)$ and $\left\{\partial_u^\dagger B\right\}(u^\dagger)$, that were introduced in (1.22) to make the adjoint approach possible, can be regarded as sensitivities to a model perturbation δm. In case of model perturbations δm with a very local influence, *i.e.* $\{\partial_m F\}(\delta m) = \delta(\boldsymbol{x}-\boldsymbol{y})$, $\{\partial_m B\}(\delta m) = \delta(\boldsymbol{x}-\boldsymbol{y})$ or $\{\partial_m C\}(\delta m) = \delta(\boldsymbol{x}-\boldsymbol{y})$, the Gâteaux derivative of J would be $\delta J = u^\dagger(\boldsymbol{y})$, $\delta J = \{\{\partial_u C\}(u^\dagger)\}(\boldsymbol{y})$ or $\delta J = \{\{\partial_u B\}(u^\dagger)\}(\boldsymbol{y})$, which could indeed be interpreted as the value of sensitivities at the point of the localized perturbation. However, in contrast to h' in (1.17), which represents a sensitivity to a state space perturbation δu expressed in the corresponding model space $U_2(\Omega)$, the sensitivities $u^\dagger$, $\left\{\partial_u^\dagger C\right\}(u^\dagger)$ and $\left\{\partial_u^\dagger B\right\}(u^\dagger)$ are not expressed in the space corresponding to the perturbation they represent, *i.e.* they are not expressed in the model space M. This useful property can however be obtained at the cost of a second dualization. One shall for this work with the adjoint operator of $\partial_m F(u,m)$, which is defined by the identity

$$\left\langle u^\dagger, \{\partial_m F\}(\delta m)\right\rangle_{U_2^\dagger(\Omega)} = \left\langle \left\{\partial_m^\dagger F\right\}(u^\dagger), \delta m\right\rangle_{M_2(\Omega)} + \left[u^\dagger, \delta m\right]_{\partial_m F}. \tag{1.29}$$

Similar to (1.23), the boundary term $\left[u^\dagger, \delta m\right]_{\partial_m F}$ is eliminated by an appropriate definition of adjoint trace-based operators

$$\left\{\partial_m \tilde{B}(u,m)\right\}(\cdot) : M \to C_2(\partial\Omega) \quad \text{and} \quad \left\{\partial_m \tilde{C}(u,m)\right\}(\cdot) : M \to B_2(\partial\Omega) \tag{1.30}$$

such that the boundary term verifies the identity

$$\begin{aligned}\left[u^\dagger, \delta m\right]_{\partial_m F} = &\left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m \tilde{C}\right\}(\delta m) - \{\partial_m C\}(\delta m)\right\rangle_{C_2(\partial\Omega)} \\ &- \left\langle \{\partial_u C\}(u^\dagger), \left\{\partial_m \tilde{B}\right\}(\delta m) - \{\partial_m B\}(\delta m)\right\rangle_{B_2(\partial\Omega)}.\end{aligned} \tag{1.31}$$

It is interesting to rearrange the terms of the previous equation as follows

$$\begin{aligned}&\left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \{\partial_m B\}(\delta m)\right\rangle_{B_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \{\partial_m C\}(\delta m)\right\rangle_{C_2(\partial\Omega)} + \left[u^\dagger, \delta m\right]_{\partial_m F} \\ &\quad = \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \left\{\partial_m \tilde{B}\right\}(\delta m)\right\rangle_{B_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m \tilde{C}\right\}(\delta m)\right\rangle_{C_2(\partial\Omega)}\end{aligned} \tag{1.32}$$

to highlight how the original trace-based operators $\partial_m B$ and $\partial_m C$ have been slightly modified to assimilate the boundary term $\left[u^\dagger, \delta m\right]_{\partial_m F}$ introduced by the second dualization, after which

the Gâteaux derivative of J reads

$$\delta J = -\left\langle \left\{\partial_m^\dagger F\right\}(u^\dagger), \delta m\right\rangle_{M_2(\Omega)} - \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \left\{\partial_m \tilde{B}\right\}(\delta m)\right\rangle_{B_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m \tilde{C}\right\}(\delta m)\right\rangle_{C_2(\partial\Omega)}. \quad (1.33)$$

A last step is needed to complete this adjoint theoretical framework with boundary perturbations. The boundary operators $\partial_m \tilde{B}$ and $\partial_m \tilde{C}$ in (1.33) depend on u and are therefore not suited for the definition of a sensitivity in the model space M. Again, it would be convenient to move the operators from the second to the first slot of the scalar products by means of a further dualization in order to isolate δm in the second slot in both terms. As these terms are surface terms defined on $\partial\Omega$, the dualization is straightforward provided the operators are true surface differential operators. Indeed, the definition of the adjoint operator entails no boundary term in that case because, $\partial\Omega$ being a boundary, it has no boundary itself, *i.e.* $\partial\partial\Omega = 0$.
In many cases in practice, without loss of generality, the boundary trace-based operators $\partial_m \tilde{B}$ and $\partial_m \tilde{C}$ can be written as the composition of a trace $P : M(\Omega) \to P_2(\partial\Omega)$ independent of u and m, and purely boundary differential operators $\left\{\partial_m \tilde{B}_P(u,m)\right\}(\cdot) : P_2(\partial\Omega) \to B_2(\partial\Omega)$ and $\left\{\partial_m \tilde{C}_P(u,m)\right\}(\cdot) : P_2(\Omega) \to C_2(\partial\Omega)$ so that the last two term in (1.33) can be rewritten

$$\left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \left\{\partial_m \tilde{B}_P\right\}(P(\delta m))\right\rangle_{B_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \left\{\partial_m \tilde{C}_P\right\}(P(\delta m))\right\rangle_{C_2(\partial\Omega)} \quad (1.34)$$

and, equivalently, in terms of the corresponding adjoint operators,

$$\left\langle \left\{\partial_m^\dagger \tilde{B}_P\right\}\left(\left\{\partial_u^\dagger C\right\}(u^\dagger)\right), P(\delta m)\right\rangle_{P_2(\partial\Omega)} - \left\langle \left\{\partial_m^\dagger \tilde{C}_P\right\}\left(\left\{\partial_u^\dagger B\right\}(u^\dagger)\right), P(\delta m)\right\rangle_{P_2(\partial\Omega)}. \quad (1.35)$$

The Gâteaux derivative of J finally writes

$$\begin{aligned}\delta J = &-\left\langle \left\{\partial_m^\dagger F\right\}(u^\dagger), \delta m\right\rangle_{M_2(\Omega)} \\ &- \left\langle \left\{\partial_m^\dagger \tilde{B}_P\right\}\left(\left\{\partial_u^\dagger C\right\}(u^\dagger)\right) - \left\{\partial_m^\dagger \tilde{C}_P\right\}\left(\left\{\partial_u^\dagger B\right\}(u^\dagger)\right), P(\delta m)\right\rangle_{P_2(\partial\Omega)} \quad (1.36)\\ := &\ \langle j'_\Omega, \delta m\rangle_{M_2(\Omega)} \\ &+ \langle j'_{\partial\Omega}, P(\delta m)\rangle_{P_2(\partial\Omega)}, \quad (1.37)\end{aligned}$$

with the *bulk sensitivity* to model perturbations

$$j'_\Omega := -\left\{\partial_m^\dagger F\right\}(u^\dagger), \quad (1.38)$$

and the *boundary sensitivity* to model trace perturbations

$$j'_{\partial\Omega} := \left\{\partial_m^\dagger \tilde{B}_P\right\}\left(\left\{\partial_u^\dagger C\right\}(u^\dagger)\right) - \left\{\partial_m^\dagger \tilde{C}_P\right\}\left(\left\{\partial_u^\dagger B\right\}(u^\dagger)\right). \tag{1.39}$$

These kernels gradients have the same interpretation than h' and k' as sensitivities, but now for model perturbations δm rather than state perturbations δu.
The classical result

$$\{D_m J\}(\delta m) = \langle j'_\Omega, \delta m\rangle_{M_2(\Omega)} \tag{1.40}$$

is obtained either when the trace $P(\delta m)$ vanishes, or when the boundary gradient $j'_{\partial\Omega}$ vanishes. When the model parameter only lives on the boundary $\partial\Omega$, it is reasonable to consider that the direct operator F does not depend on m. Moreover the boundary operators $\partial_m B : M(\partial\Omega) \to B_2(\partial\Omega)$ and $\partial_m C : M(\partial\Omega) \to C_2(\partial\Omega)$ are in this case *purely boundary operators*, as the model parameter space is a boundary function space. In this particular case, the directional derivative (1.28) can be expressed in the model space as

$$\{D_m J\}(\delta m) = -\left\langle \left\{\partial_m^\dagger B\right\}\left(\left\{\partial_u^\dagger C\right\}(u^\dagger)\right) - \left\{\partial_m^\dagger C\right\}\left(\left\{\partial_u^\dagger B\right\}(u^\dagger)\right), \delta m\right\rangle_{M_2(\partial\Omega)} \tag{1.41}$$

$$:= \langle j'_{\partial\Omega}, \delta m\rangle_{M_2(\partial\Omega)}. \tag{1.42}$$

1.4.4 Hessian kernels

The adjoint state method described here above does not enable the efficient computation of the entire Hessian operator. Nevertheless, it enables to compute efficiently the application of this Hessian operator in a specific direction δm. Indeed, the Hessian operator in the direction δm is defined as the directional derivative of the gradient, *cf.* (1.11). Because the gradient is expressed in terms of the forward state u and the adjoint state $u^\dagger$ through (1.38) and (1.39), its directional derivative in a direction δm is expressed in terms of the perturbed forward state δu and the perturbed adjoint state $\delta u^\dagger$, which can be computed through the perturbed forward system (1.13) and through the perturbed adjoint system, *i.e.* the total derivative w.r.t. m of (1.24). Consequently, for each new direction in which the application of the Hessian operator must be computed, two supplementary linear systems must be solved and for that specific reason, the Hessian operator should be used sparingly.

1.5 Time-harmonic wave scattering problems

The theoretical concepts introduced so far have been kept general. A methodology to apply them to specific boundary problems is now presented in this section. In order to cover a broad spectrum of situations, scattering problems in three different physics are treated in detail. This section is then followed by a numerical illustration in Section 1.6.

Elastodynamic, electromagnetic and acoustic time-harmonic wave propagation problems obey respectively *Navier's* [18, 182], *Maxwell's* [74, 96, 129, 5, 42, 145] or *Helmholtz's* [148, 80, 153, 119, 93, 104, 23] equations. Whereas Navier's equations are specific to elastodynamic problems and Maxwell's equations to electromagnetic problems, the Helmholtz equation (variant I) is more generic and could be used to cover problems in any of the three physics, under some assumptions. Without simplifying assumptions, acoustic problems should be modeled using a slight variant of the Helmholtz equations (variant II). In the time-harmonic regime at pulsation ω (with convention $+i\omega t$), a wave propagation problem can be formalized as a boundary value problem with a zeroth order space derivative term proportional to the square of the pulsation, plus a second order space derivative term. State variables, model parameters and direct operators of the three problems are given in Table 1.1. For each problem, multiple conventions are equivalent for the definition of the model parameters. In this thesis, parametrizations for which the direct operator only involves bi-linear (model parameter-wave field) terms, have systematically been chosen, as they appear as the most natural choice.

	State variable u	Model parameter m	Direct operator $F(u, m)$
Helmholtz I	*wave field* u	*slowness squared* s^2	$\operatorname{div}(\mathbf{grad}(u)) + \omega^2 s^2 u$
Helmholtz II	*pressure field* p	*specific volume* ν *compressibility* K	$\operatorname{div}(\nu\,\mathbf{grad}(p)) + \omega^2 K p$
Maxwell	*electric field* $\boldsymbol{e}$ *magnetic field* $\boldsymbol{h}$	*permittivity* ϵ *permeability* μ	$\begin{pmatrix} -i\omega\epsilon & \mathbf{curl}() \\ \mathbf{curl}() & i\omega\mu \end{pmatrix} \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{h} \end{pmatrix}$
Navier	*displacement field* $\boldsymbol{u}$	*1st Lamé parameter* λ *2nd Lamé parameter* μ *density* ρ	$\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) + \omega^2 \rho \boldsymbol{u}$ with $\boldsymbol{\sigma} = \lambda \operatorname{tr}\boldsymbol{\epsilon}\boldsymbol{I} + 2\mu\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon} = \frac{1}{2}(\mathbf{grad}(\boldsymbol{u}) + \mathbf{grad}^T(\boldsymbol{u}))$

Table 1.1: States variables, model parameters and direct state operators for Helmholtz's, Maxwell's and Navier's equations in the time-harmonic regime at pulsation ω (with convention $+i\omega t$).

The direct operators listed in Table 1.1 are all linear in u, *i.e.* $\{\partial_u F(u, m)\}(\delta u) = F(\delta u, m)$, and hermitian in u, *i.e.* $\partial_u^\dagger F = \overline{\partial_u F}$, where the upper bar denotes complex conjugation. The adjoint operators $\partial_u^\dagger F$, defined by (1.18), are obtained by making successively two integrations by parts (*cf.* Appendix A.1). On the other hand, the direct operators in Table 1.1 are not all linear in m, *i.e.* $\{\partial_m F(u, m)\}(\delta m) \neq F(u, \delta m)$, and can also not be hermitian as their co-domain is in general different from their domain. The m-adjoint operators $\partial_m^\dagger F$ are given in Table 1.2. For Helmholtz's and Maxwell's case, they are obtained without integration by parts,

as the direct operator has no spatial derivative of m. In Navier's case, a single integration by parts is sufficient, due to the first order spatial derivative of the Lamé parameters λ and μ that appears in the operator (*cf.* Appendix A.2).

	Model parameter m		Adjoint operator $\left\{\partial_m^\dagger F\right\}(u^\dagger)$
Helmholtz I	*slowness squared*	s^2	$\omega^2\,\overline{u}u^\dagger$
Helmholtz II	*specific volume*	ν	$-\mathbf{grad}\,(\overline{p})\cdot\mathbf{grad}\left(p^\dagger\right)$
	compressibility	K	$\omega^2\,\overline{p}p^\dagger$
Maxwell	*permittivity*	ϵ	$i\omega\,\overline{\boldsymbol{e}}\cdot\boldsymbol{e}^\dagger$
	permeability	μ	$-i\omega\,\overline{\boldsymbol{h}}\cdot\boldsymbol{h}^\dagger$
Navier	*1st Lamé parameter*	λ	$-\operatorname{div}(\overline{\boldsymbol{u}})\operatorname{div}\left(\boldsymbol{u}^\dagger\right)$
	2nd Lamé parameter	μ	$-2\,\boldsymbol{\epsilon}(\overline{\boldsymbol{u}}):\boldsymbol{\epsilon}(\boldsymbol{u}^\dagger)$
	density	ρ	$\omega^2\,\overline{\boldsymbol{u}}\cdot\boldsymbol{u}^\dagger$

Table 1.2: Adjoint w.r.t. the model parameter m of the operators given in Table 1.1.

It is interesting to note that the m-adjoint operators $\left\{\partial_m^\dagger F\right\}(u^\dagger)$ (and thus the bulk sensitivities j'_Ω, as of (1.38)) are, up to a constant factor, always the product of the direct and the adjoint fields. This observation also justifies the parametrization convention of this thesis: any other choice would lengthen the expression of the bulk sensitivity, because of the Jacobian coming from the change of variable. As mentioned earlier, the adjoint field $u^\dagger(\boldsymbol{y})$ carries information about the effect on J of a normalized local perturbation at $\boldsymbol{y}$ (*i.e.* $\{\partial_m F\}(\delta m) = \delta(\boldsymbol{x}-\boldsymbol{y})$ implies $\delta J = u^\dagger(\boldsymbol{y})$), whereas the field $u(\boldsymbol{y})$ is the actual magnitude of the corresponding state perturbation. The functional J is therefore sensitive to a model perturbation δm at $\boldsymbol{x}$ if both $u(\boldsymbol{y})$ and $u^\dagger(\boldsymbol{y})$ are sufficiently large. The fact that bulk sensitivities are linear in u directly follows from the fact that F, and thus $\partial_m F$, are linear in u.

1.5.1 Direct and adjoint boundary conditions

The derivation of the adjoint operators $\partial_u^\dagger F$ (*cf.* Appendix A.1) show that the boundary term $\left[u^\dagger, \delta u\right]_{\partial_u F}$ has the same structure for all three wave equations

$$\left[u^\dagger, \delta u\right]_{\partial_u F} = \int_{\partial\Omega} \begin{pmatrix} T_1(\overline{u}^\dagger) & T_0(\overline{u}^\dagger) \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} T_1(\delta u) \\ T_0(\delta u) \end{pmatrix} d\partial\Omega$$

where the trace T_1 involves a first order spatial derivatives, whereas the trace T_0 does not.

In Maxwell's case, there exist two equivalent ways to express the boundary term, and hence two definitions for the traces, according to whether $\boldsymbol{h}$ is regarded as a spatial derivative of $\boldsymbol{e}$ ($\boldsymbol{e}$-formulation) or the opposite ($\boldsymbol{h}$-formulation). Navier's equations yield two zeroth order traces orthogonal to each other, so that the boundary term can be split into two parts that can be studied separately. The analytic expressions of these traces are listed in Table 1.3 in terms of the standard geometric boundary operators defined as follows: the normal derivative $\frac{\partial}{\partial n} := \hat{\boldsymbol{n}} \cdot \mathbf{grad}\,()$, the normal and tangential components, $\gamma_n() := \hat{\boldsymbol{n}}\,(\hat{\boldsymbol{n}}\cdot)$ and $\gamma_T() := -\hat{\boldsymbol{n}} \times (\hat{\boldsymbol{n}}\times)$, $\gamma_n() + \gamma_T() = 1$, and finally the orthogonal tangential component $\gamma_t() := \hat{\boldsymbol{n}}\times$.

	$T_1(u, m)$	$T_0(u)$
Helmholtz I	$\frac{\partial u}{\partial n}$	u
Helmholtz II	$\nu \frac{\partial p}{\partial n}$	p
Maxwell(-$\boldsymbol{e}$)	$\gamma_t(\boldsymbol{h})$	$\gamma_T(\boldsymbol{e})$
Maxwell(-$\boldsymbol{h}$)	$\gamma_t(\boldsymbol{e})$	$\gamma_T(\boldsymbol{h})$
Navier	$\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}}$	$\gamma_T(\boldsymbol{u})$ and $\gamma_n(\boldsymbol{u})$

Table 1.3: First order trace T_1 and zeroth order trace T_0 appearing in the boundary term $\left[u^\dagger, \delta u\right]_{\partial_u F}$ for Helmholtz's, Maxwell's and Navier's equations in time-harmonic regime.

The direct, adjoint, m-adjoint and boundary operators being now determined for the three wave propagation problems at hand, one can proceed and establish the *adjoint problem with boundary perturbation* (1.24). It is first noted that the boundary perturbations $\partial_u B$ and $\partial_u C$ are dictated by the physics of the considered problem, and it is customary that they can be expressed as a linear combination of boundary operators T_0 and T_1 obtained above. The first step to establish (1.24) is to find a pair of adjoint boundary operators $\partial_u^\dagger B$ and $\partial_u^\dagger C$ that satisfy (1.23). To that purpose, it is also natural to look for adjoint operators that are linear combinations of T_0 and T_1 [63], so that one shall write altogether

$$\partial_u B = b_1\, T_1 + b_0\, T_0 \quad \text{and} \quad \partial_u B^\dagger = b_1^\dagger\, \overline{T}_1 + b_0^\dagger\, \overline{T}_0, \tag{1.43}$$

$$\partial_u C = c_1\, T_1 + c_0\, T_0 \quad \text{and} \quad \partial_u C^\dagger = c_1^\dagger\, \overline{T}_1 + c_0^\dagger\, \overline{T}_0. \tag{1.44}$$

These direct and adjoint boundary conditions thus depend on the model parameter m not only through the $b_k^{(\dagger)}(m)$ and $c_k^{(\dagger)}(m)$ coefficients, but also through the first order trace T_1 (*cf.* Table 1.3).

Using a matrix formalism

$$\boldsymbol{T}(u) := \begin{pmatrix} T_1(u) & T_0(u) \end{pmatrix}^T, \qquad \boldsymbol{b}^{(\dagger)} := \begin{pmatrix} b_1^{(\dagger)} & b_0^{(\dagger)} \end{pmatrix}^T \quad \text{and} \quad \boldsymbol{c}^{(\dagger)} := \begin{pmatrix} c_1^{(\dagger)} & c_0^{(\dagger)} \end{pmatrix}^T, \tag{1.45}$$

they can be written compactly

$$\{\partial_u B\}(\delta u) = \boldsymbol{b}^T \boldsymbol{T}(\delta u) \quad \text{and} \quad \left\{\partial_u^\dagger \overline{B}\right\}(\overline{u}^\dagger) = \boldsymbol{T}^T(\overline{u}^\dagger)\overline{\boldsymbol{b}}^\dagger, \tag{1.46}$$

$$\{\partial_u C\}(\delta u) = \boldsymbol{c}^T \boldsymbol{T}(\delta u) \quad \text{and} \quad \left\{\partial_u^\dagger \overline{C}\right\}(\overline{u}^\dagger) = \boldsymbol{T}^T(\overline{u}^\dagger)\overline{\boldsymbol{c}}^\dagger \tag{1.47}$$

and the boundary term as

$$\left[u^\dagger, \delta u\right]_{\partial_u F} = \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger)\boldsymbol{A}\, \boldsymbol{T}(\delta u)\, d\partial\Omega \tag{1.48}$$

with

$$\boldsymbol{A} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{1.49}$$

Condition (1.23) now reads

$$\begin{aligned} &\left[u^\dagger, \delta u\right]_{\partial_u F} \\ &\quad = \left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \{\partial_u C\}(\delta u)\right\rangle_{C_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \{\partial_u B\}(\delta u)\right\rangle_{B_2(\partial\Omega)} \end{aligned} \tag{1.50}$$

$$\begin{aligned} \Leftrightarrow &\int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger)\boldsymbol{A}\, \boldsymbol{T}(\delta u)\, d\partial\Omega \\ &\quad = \int_{\partial\Omega} \left(\boldsymbol{T}^T(\overline{u}^\dagger)\overline{\boldsymbol{b}}^\dagger\right)\left(\boldsymbol{c}^T\boldsymbol{T}(\delta u)\right) - \left(\boldsymbol{T}^T(\overline{u}^\dagger)\overline{\boldsymbol{c}}^\dagger\right)\left(\boldsymbol{b}^T\boldsymbol{T}(\delta u)\right)\, d\partial\Omega \qquad (1.51) \\ &\quad = \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger)\left(\overline{\boldsymbol{b}}^\dagger\boldsymbol{c}^T - \overline{\boldsymbol{c}}^\dagger\boldsymbol{b}^T\right)\boldsymbol{T}(\delta u)\, d\partial\Omega \qquad (1.52) \end{aligned}$$

which reduces to a simple matrix equation

$$\boldsymbol{A} = \overline{\boldsymbol{b}}^\dagger\boldsymbol{c}^T - \overline{\boldsymbol{c}}^\dagger\boldsymbol{b}^T \tag{1.53}$$

or explicitly

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \overline{b}_1^\dagger c_1 - \overline{c}_1^\dagger b_1 & \overline{b}_1^\dagger c_0 - \overline{c}_1^\dagger b_0 \\ \overline{b}_0^\dagger c_1 - \overline{c}_0^\dagger b_1 & \overline{b}_0^\dagger c_0 - \overline{c}_0^\dagger b_0 \end{pmatrix}. \tag{1.54}$$

Specified boundary performance The operators $\partial_u B$ and $\partial_u C$ are in general known from the problem statement. Equation (1.54) can then be solved to express the coefficients of the adjoint operators in terms of the known coefficients of the direct operators [63]

$$\begin{pmatrix} \overline{c}_1^\dagger & \overline{c}_0^\dagger \\ \overline{b}_1^\dagger & \overline{b}_0^\dagger \end{pmatrix} = \frac{1}{b_0 c_1 - b_1 c_0}\begin{pmatrix} c_1 & c_0 \\ b_1 & b_0 \end{pmatrix} \tag{1.55}$$

so that the adjoint operators are

$$\partial_u^\dagger \overline{B} = \frac{1}{b_0 c_1 - b_1 c_0}\partial_u B \quad \text{and} \quad \partial_u^\dagger \overline{C} = \frac{1}{b_0 c_1 - b_1 c_0}\partial_u C, \tag{1.56}$$

completing so the formulation of the *adjoint problem with boundary perturbation* (1.24) and the formulation of the Gâteaux derivative of J (1.33).
In the particular cases considered here, further simplifications can be done that follow from the fact that the operators F and B are hermitian and linear in u. One has thus successively

$$\overline{h}' = \left\{\overline{\partial_u^\dagger F}(u,m)\right\}(\overline{u}^\dagger) = \{\partial_u F(u,m)\}(\overline{u}^\dagger) = F(\overline{u}^\dagger, m)$$

and similarly for B

$$\overline{k}' = \left\{\overline{\partial_u^\dagger B}(u,m)\right\}(\overline{u}^\dagger) = \frac{1}{b_0c_1 - b_1c_0}\{\partial_u B(u,m)\}(\overline{u}^\dagger) = \frac{1}{b_0c_1 - b_1c_0}B(\overline{u}^\dagger, m),$$

so that the adjoint problem (1.24) can remarkably be written in terms of the direct operators

$$\begin{cases} F(\overline{u}^\dagger, m) &= \overline{h}', \\ B(\overline{u}^\dagger, m) &= (b_0c_1 - b_1c_0)\overline{k}'. \end{cases} \tag{1.57}$$

Whenever the boundary trace-based operators are proportional to each other, *i.e.* $\partial_u C \propto \partial_u B$, one has $b_0c_1 - b_1c_0 = 0$ and the system (1.54) cannot be solved. The degeneracy comes from the fact that the observed operator $\partial_u C$ is actually proportional to the imposed condition $\partial_u B = -\partial_m B$ (*cf.* (1.13)). The observed operator does therefore not depend on the state perturbation δu, and the definition of an adjoint state on such regions is useless. The degeneracy is resolved by substituting $\partial_u B = -\partial_m B$ to $\partial_u C$ in (1.17), and then by following the procedure as if there were no boundary performance functional $\partial_u C$ defined on these regions (*cf.* paragraph below).

Unspecified boundary performance When the performance functional J has no boundary term, the operator $\partial_u C$ is not defined. The system (1.54) is then under-determined and a supplementary condition can be imposed arbitrarily. A convenient condition is

$$b_0c_1 - c_0b_1 = 1 \tag{1.58}$$

since (1.54) then yields

$$\partial_u^\dagger B = \partial_u \overline{B} \quad \text{and} \quad \partial_u^\dagger C = \partial_u \overline{C} \tag{1.59}$$

which also leads to an adjoint problem (1.24) written in terms of the direct operators.

1.5.2 Boundary sensitivity to model parameters

The second step to establish (1.24) is the derivation of the operators $\partial_m B$ and $\partial_m C$ appearing in the directional derivative (1.28). With the same matrix formalism as above, they read

$$\{\partial_m B\}(\delta m) = \boldsymbol{\delta b}^T \boldsymbol{T}(u) + \boldsymbol{b}^T \{\partial_m \boldsymbol{T}(u)\}(\delta m) \tag{1.60}$$

and

$$\{\partial_m C\}(\delta m) = \boldsymbol{\delta c}^T \boldsymbol{T}(u) + \boldsymbol{c}^T \{\partial_m \boldsymbol{T}(u)\}(\delta m), \tag{1.61}$$

and the two boundary terms of (1.28) can be successively modified as follows

$$\left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \{\partial_m C\}(\delta m)\right\rangle_{C_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \{\partial_m B\}(\delta m)\right\rangle_{B_2(\partial\Omega)} \tag{1.62}$$

$$= \int_{\partial\Omega} \left(\boldsymbol{T}^T(\overline{u}^\dagger)\overline{\boldsymbol{b}}^\dagger\right) \left(\boldsymbol{\delta c}^T \boldsymbol{T}(u) + \boldsymbol{c}^T \{\partial_m \boldsymbol{T}(u)\}(\delta m)\right) d\partial\Omega - \int_{\partial\Omega} \left(\boldsymbol{T}^T(\overline{u}^\dagger)\overline{\boldsymbol{c}}^\dagger\right) \left(\boldsymbol{\delta b}^T \boldsymbol{T}(u) + \boldsymbol{b}^T \{\partial_m \boldsymbol{T}(u)\}(\delta m)\right) d\partial\Omega \tag{1.63}$$

$$= \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger) \left(\overline{\boldsymbol{b}}^\dagger \boldsymbol{\delta c}^T - \overline{\boldsymbol{c}}^\dagger \boldsymbol{\delta b}^T\right) \boldsymbol{T}(u)\, d\partial\Omega + \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger) \left(\overline{\boldsymbol{b}}^\dagger \boldsymbol{c}^T - \overline{\boldsymbol{c}}^\dagger \boldsymbol{b}^T\right) \{\partial_m \boldsymbol{T}(u)\}(\delta m)\, d\partial\Omega \tag{1.64}$$

$$= \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger) \left(\overline{\boldsymbol{b}}^\dagger \boldsymbol{\delta c}^T - \overline{\boldsymbol{c}}^\dagger \boldsymbol{\delta b}^T\right) \boldsymbol{T}(u)\, d\partial\Omega + \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger)\, \boldsymbol{A}\, \{\partial_m \boldsymbol{T}(u)\}(\delta m)\, d\partial\Omega. \tag{1.65}$$

The boundary term $\left[u^\dagger, \delta m\right]_{\partial_m F}$ in (1.29), on the other hand, vanishes in Maxwell's and Helmholtz I's cases whereas in Navier's and Helmholtz II's cases, it can be written (*cf.* Appendix A.2)

$$\left[u^\dagger, \delta m\right]_{\partial_m F} = \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger)\, \boldsymbol{A}\, \{\partial_m \boldsymbol{T}(u)\}(\delta m)\, d\partial\Omega.$$

Summing up all boundary terms of (1.28), which are also the left-hand side of (1.32), one obtains

$$\left\langle \left\{\partial_u^\dagger B\right\}(u^\dagger), \{\partial_m C\}(\delta m)\right\rangle_{C_2(\partial\Omega)} - \left\langle \left\{\partial_u^\dagger C\right\}(u^\dagger), \{\partial_m B\}(\delta m)\right\rangle_{B_2(\partial\Omega)} - \left[u^\dagger, \delta m\right]_{\partial_m F} = \int_{\partial\Omega} \boldsymbol{T}^T(\overline{u}^\dagger) \left(\overline{\boldsymbol{b}}^\dagger \boldsymbol{\delta c}^T - \overline{\boldsymbol{c}}^\dagger \boldsymbol{\delta b}^T\right) \boldsymbol{T}(u)\, d\partial\Omega, \tag{1.66}$$

where it is to note that the terms in δm have canceled out. The modified boundary operators can thus be identified using (1.32), and they are simply given by

$$\left\{\partial_m \tilde{B}\right\}(\delta m) = \boldsymbol{\delta b}^T \boldsymbol{T}(u) \quad \text{and} \quad \left\{\partial_m \tilde{C}\right\}(\delta m) = \boldsymbol{\delta c}^T \boldsymbol{T}(u). \tag{1.67}$$

The coefficients $b_0(m)$, $c_0(m)$ and $b_1(m)$, $c_1(m)$ are usually simple functions, *i.e.* they involve no derivative on m. Their perturbation are then proportional to δm

$$\boldsymbol{\delta b} = \frac{\partial \boldsymbol{b}}{\partial m}\delta m \quad \text{and} \quad \boldsymbol{\delta c} = \frac{\partial \boldsymbol{c}}{\partial m}\delta m \tag{1.68}$$

and the boundary sensitivity to model perturbation (1.39) is obtained by factorization

$$\vec{j}'_{\partial\Omega} = \boldsymbol{T}^T(\overline{u}^\dagger)\left(\overline{\boldsymbol{b}}^\dagger\frac{\partial \boldsymbol{c}^T}{\partial m} - \overline{\boldsymbol{c}}^\dagger\frac{\partial \boldsymbol{b}^T}{\partial m}\right)\boldsymbol{T}(u) \tag{1.69}$$

$$= \frac{1}{b_0c_1 - b_1c_0}\boldsymbol{T}^T(\overline{u}^\dagger)\left(\boldsymbol{b}\frac{\partial \boldsymbol{c}^T}{\partial m} - \boldsymbol{c}\frac{\partial \boldsymbol{b}^T}{\partial m}\right)\boldsymbol{T}(u), \tag{1.70}$$

or explicitly

$$\vec{j}'_{\partial\Omega} = \frac{1}{b_0c_1 - b_1c_0}\boldsymbol{T}^T(\overline{u}^\dagger)\begin{pmatrix} b_1\frac{\partial c_1}{\partial m} - c_1\frac{\partial b_1}{\partial m} & b_1\frac{\partial c_0}{\partial m} - c_1\frac{\partial b_0}{\partial m} \\ b_0\frac{\partial c_1}{\partial m} - c_0\frac{\partial b_1}{\partial m} & b_0\frac{\partial c_0}{\partial m} - c_0\frac{\partial b_0}{\partial m}. \end{pmatrix}\boldsymbol{T}(u). \tag{1.71}$$

Similar to the bulk sensitivity $\vec{j}'_\Omega$ (*cf.* Table 1.2), the boundary sensitivity $\vec{j}'_{\partial\Omega}$ is proportional to the product of traces of the direct and the adjoint fields. Examples of boundary sensitivities $j'_{\partial\Omega}$ corresponding to particular pairs of direct and performance boundary trace-based operators B and C are listed in Table 1.4. The chosen examples correspond to specific physically or mathematically grounded choices for the coefficients $b_1(m)$, $b_0(m)$, $c_1(m)$ and $c_0(m)$, which cover a large range of the situations encountered in practical modeling problems.

Model parameter	m	b_1	b_0	c_1	c_0	$j'_{\partial\Omega}$
Impedance	z	1	z	0	-1	$+\ T_0(u)T_0(\overline{u}^\dagger)$
Admittance	y	y	1	1	0	$-\ T_1(u)T_1(\overline{u}^\dagger)$
Dirichlet	z	0	z	z^{-1}	0	$-z^{-1}\ \left[T_0(u)T_1(\overline{u}^\dagger) + T_1(u)T_0(\overline{u}^\dagger)\right]$
Neumann	y	y	0	0	$-y^{-1}$	$y^{-1}\ \left[T_0(u)T_1(\overline{u}^\dagger) + T_1(u)T_0(\overline{u}^\dagger)\right]$
Mixed	α	α	α	$(2\alpha)^{-1}$	$-(2\alpha)^{-1}$	$\alpha^{-1}\ \left[T_0(u)T_0(\overline{u}^\dagger) - T_1(u)T_1(\overline{u}^\dagger)\right]$

Table 1.4: Boundary sensitivities $j'_{\partial\Omega}$ for particular pairs of direct and performance boundary trace-based operators $B = b_1T_1 + b_0T_0$ and $C = c_1T_1 + c_0T_0$, *i.e.* for given expressions of the coefficients $b_1(m)$, $b_0(m)$, $c_1(m)$ and $c_0(m)$ as functions of m.

Unspecified boundary performance Whenever C is not defined, it can be freely chosen. In the previous subsection, the additional condition $b_0c_1 - c_0b_1 = 1$ was shown to be a natural choice. Depending on the value of b_0 and b_1, the boundary sensitivity (1.71) can be given a more or less compact form if the coefficients c_0 and c_1 are chosen appropriately on basis,

e.g., of the examples listed in Table 1.4 for some frequent boundary conditions.

1.5.3 Sobolev gradient kernel for scalar model parameters

Bulk (cf. Table 1.1) and boundary (cf. Table 1.4) model parameters have been considered in the last two sections. Both are scalar parameters and, in view of their physical meaning, their distribution can be assumed to belong to $L_2(\Omega)$ and $L_2(\partial\Omega)$ spaces respectively, or to more regular subspaces of them. This choice of an appropriate representation space for model parameters is part of the modeling and three examples are discussed below.

1. If there is no reason for a bulk model parameter m to exhibit a smooth distribution, one is led to choose $M = M_2(\Omega) = L_2(\Omega)$. Model parameter traces can then not be defined and the boundary sensitivity thus vanishes. The inner product in M is naturally chosen to be the inner product in $L_2(\Omega)$, so that one has by definition of the gradient kernel

$$\delta J = \{D_m J\}(\delta m) = \langle j', \delta m\rangle_{L_2(\Omega)}\,. \tag{1.72}$$

   Equation (1.37) then gives

$$\delta J = \langle j'_\Omega, \delta m\rangle_{M_2(\Omega)} = \langle j'_\Omega, \delta m\rangle_{L_2(\Omega)} \tag{1.73}$$

   and hence, by identification,

$$j' = j'_\Omega. \tag{1.74}$$

2. Similarly, the distribution of a not particularly smooth boundary model parameter m will be sought in $M = M_2(\partial\Omega) = L_2(\partial\Omega)$. The inner product in M is then naturally chosen as the inner product in $L_2(\partial\Omega)$, so that one has by definition of the gradient kernel

$$\delta J = \{D_m J\}(\delta m) = \langle j', \delta m\rangle_{L_2(\partial\Omega)}\,, \tag{1.75}$$

   and (1.42) gives

$$\delta J = \langle j'_{\partial\Omega}, \delta m\rangle_{M_2(\partial\Omega)} = \langle j'_{\partial\Omega}, \delta m\rangle_{L_2(\partial\Omega)} \tag{1.76}$$

   so that one ends up, by identification, with

$$j' = j'_{\partial\Omega}. \tag{1.77}$$

3. Now if the model parameter m is expected to have regularity, *e.g.* due to its physical meaning, its distribution can be sought in the space $M = H_1(\Omega)$. The inner product in

M is then chosen as the natural inner product in $H_1(\Omega)$, *i.e.*

$$\langle m_2, m_1 \rangle_M := \langle m_2, m_1 \rangle_{L_2(\Omega)} + \alpha_1 \langle \mathbf{grad}\,(m_2)\,, \mathbf{grad}\,(m_1) \rangle_{L_2^3(\Omega)}\,. \tag{1.78}$$

with α_1 a strictly positive parameter. From (1.37), one has

$$\langle j', \delta m \rangle_{L_2(\Omega)} + \alpha_1 \langle \mathbf{grad}\,(j')\,, \mathbf{grad}\,(\delta m) \rangle_{L_2^3(\Omega)} = \langle j'_\Omega, \delta m \rangle_{L_2(\Omega)} + \langle j'_{\partial\Omega}, P(\delta m) \rangle_{L_2(\partial\Omega)}\,, \tag{1.79}$$

which is the weak form of a boundary value problem that can be solved for the gradient j'. Considering that only the Dirichlet trace of δm is involved, *i.e.* $P(\delta m) = \delta m$, the Euler-Lagrange equations of this boundary value problem are

$$\begin{cases} -\alpha_1 \mathrm{div}\,(\mathbf{grad}\,(j')) + j' = j'_\Omega, \\ \alpha_1 \dfrac{\partial j'}{\partial n} = j'_{\partial\Omega}. \end{cases} \tag{1.80}$$

Solving (1.80) can be regarded as a smoothing, or more generally as a preconditioning, of the classical L_2 gradients (1.74) and (1.77). The obtained solution is called a *Sobolev* gradient kernel [42, 127, 154, 155, 210]. Depending on the constraints or the prior information available for the model parameters, other inner products can be used advantageously, as will be illustrated in the following chapters.

1.6 Numerical illustration

This section illustrates the theoretical results exposed above, in the case of the Helmholtz I equation. Corresponding results for the three other equations are given in Appendix A. The typical performance functional for full waveform inversion is the the least squares distance between the predicted wave field u and the measurements d recorded at a set of points $\boldsymbol{x}_r$

$$H(u) = \frac{1}{2} \sum_r |u(\boldsymbol{x}_r) - d_r|^2\,. \tag{1.81}$$

This performance functional does however not integrate enough information in practice, and multiple emitters are often considered successively. The performance functional then becomes

$$J(m) = \frac{1}{2} \sum_e \sum_r |u_e(\boldsymbol{x}_r) - d_{e,r}|^2 \tag{1.82}$$

where u_e denotes the direct state field associated with the emitter e. Note that the subscript '0' for the measured data is omitted for this chapter, for the sake of compactness.
The geometrical setting of this application example consists of a cylinder-shaped inclusion Ω_c embedded in a square background medium Ω_0 with emitter/receiver arrays disposed on both

sides, as depicted in Figure 1.1. The bulk model property in this problem is the slowness squared s^2, which is by definition the squared inverse of the wave speed.

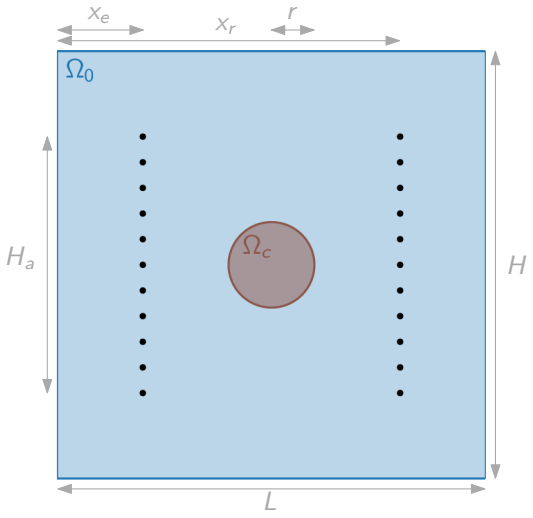


Figure 1.1: Geometrical setting of this application example: $n_e = 11$, $x_e = 5$, $H_a = 15$, $n_r = 11$, $x_r = 20$, $H = 25$, $L = 25$, $r = 2.5$. The pulsation ω is set to 2π, so that the reference wavelength is $\tilde{\lambda}_0 = 1/\tilde{s}_0 = 1$. The background and the cylinder domain overlap, *i.e.* $\Omega_0 \cap \Omega_c = \Omega_c$.

The direct wave field u_e is caused by point source emitters located at $\boldsymbol{x}_e \in \Omega_0$, and is the solution of the partial differential equation

$$\operatorname{div}\left(\mathbf{grad}\left(u_e\right)\right) + \omega^2 s^2 u_e = \delta(\boldsymbol{x} - \boldsymbol{x}_e). \tag{1.83}$$

The unbounded propagation domain is modeled by means of an absorbing boundary condition

$$\frac{\partial u_e}{\partial n} + \beta_0(s^2) u_e = 0 \tag{1.84}$$

imposed on the boundary $\partial\Omega_0$ of the computational domain. For simplicity, a zeroth order absorbing condition has been chosen, for which the relationship $\beta_0(s^2) := i\omega\sqrt{s^2}$ holds. The scattering cylinder can be modeled in two different ways. If the cylinder is made of a highly penetrable material, the propagation equation is solved explicitly inside the cylinder, and the computational domain is $\Omega = \Omega_0$. If on the other hand the cylinder is made of a weakly penetrable material, it is represented by means of an impedance boundary condition

$$\frac{\partial u_e}{\partial n} + \beta_c u_e = 0 \tag{1.85}$$

imposed on the boundary $\partial\Omega_c$ of the cylinder, and the computational domain is then $\Omega = \Omega_0 \setminus \Omega_c$.

The distribution of bulk model parameters s_c^2 in the cylinder and/or of s_0^2 in the background medium are unknown, as well as the distribution of the boundary material parameters β_c

and β_0. The minimization of the performance functional aims at determining the model parameter distribution, both in the bulk and on the boundaries, that yields the best match of the simulated wave fields with measurements. Different modeling configurations have been considered to highlight the specific role of the bulk and the boundary terms in the evaluation of the directional derivatives and of the gradient kernels. The unknown model property is either the homogeneous squared slowness s_c^2 of the cylinder assumed penetrable in Subsection 1.6.1, the homogeneous equivalent boundary impedance β_c of the cylinder assumed impenetrable in Subsection 1.6.2, the homogeneous squared slowness s_0^2 of the background medium knowing all cylinder properties in Subsection 1.6.3, or finally, the squared slowness distribution $s_0^2(\boldsymbol{x})$ of the background space with no *a priori* knowledge of the geometry of the scattering inclusion in Subsection 1.6.4.

In this academic illustration of the theory, real measurement data is not available. The measurement values used in the performance functional J are obtained from the simulation of a reference problem, with specific values of the material parameters, called ground-truth values and denoted with a tilde symbol. In all considered examples, the systematic steps of the adjoint state method are as follows, in terms of a computational domain Ω, a bulk model parameter s^2 and a boundary model parameter β that are defined case by case:

1. Find the direct states u_e obeying the direct problem

$$\begin{cases} \operatorname{div}\left(\mathbf{grad}\left(u_e\right)\right) + \omega^2 s^2 u_e &= \delta(\boldsymbol{x} - \boldsymbol{x}_e) \quad \text{in } \Omega \\ \dfrac{\partial u_e}{\partial n} + \beta u_e &= 0 \qquad\qquad \text{on } \partial\Omega. \end{cases} \tag{1.86}$$

2. Find the adjoint states $u_e^\dagger$ obeying the adjoint problem (1.24)

$$\begin{cases} \operatorname{div}\left(\mathbf{grad}\left(\overline{u}_e^\dagger\right)\right) + \omega^2 s^2 \overline{u}_e^\dagger &= \displaystyle\sum_r (\overline{u}_e(\boldsymbol{x}_r) - \overline{d}_{e,r})\delta(\boldsymbol{x} - \boldsymbol{x}_r) \quad \text{in } \Omega \\ \dfrac{\partial \overline{u}_e^\dagger}{\partial n} + \beta \overline{u}_e^\dagger &= 0 \qquad\qquad\qquad\qquad\qquad\quad \text{on } \partial\Omega. \end{cases} \tag{1.87}$$

   The source term in the adjoint problem is the gradient kernel of the bulk performance functional (h'), whose computation is done explicitly in Appendix A.3.1.

3. Once the direct and adjoint states are known, the directional derivatives with respect to the different model parameters can be evaluated, as well as the associated gradient kernels :

   - directional derivative w.r.t. β, and associated gradient kernel by identification

$$\delta J = \left\{D_\beta J(\beta)\right\}(\delta\beta) = \sum_e \left\langle u_e^\dagger, \delta\beta\, u_e \right\rangle_{L_2(\partial\Omega)} \tag{1.88}$$

hence

$$\bar{j}' = \sum_e u_e \overline{u}_e^\dagger \tag{1.89}$$

- directional derivative w.r.t. s^2 and associate gradient kernel by identification

$$\delta J = \left\{ D_{s^2} J(s^2) \right\} (\delta s^2) = -\sum_e \left\langle u_e^\dagger, \omega^2 \, \delta s^2 \, u_e \right\rangle_{L_2(\Omega)} \tag{1.90}$$

hence

$$\bar{j}' = -\omega^2 \sum_e u_e \overline{u}_e^\dagger \tag{1.91}$$

- directional derivative w.r.t. s^2, with β a function of s^2

$$\delta J = \left\{ D_{s^2} J(s^2) \right\} (\delta s^2) = \\ -\sum_e \left\langle u_e^\dagger, \omega^2 \, \delta s^2 \, u_e \right\rangle_{L_2(\Omega)} + \sum_e \left\langle u_e^\dagger, \frac{\partial \beta}{\partial s^2} \delta s^2 \, u_e \right\rangle_{L_2(\partial\Omega)} \tag{1.92}$$

and the H_1-Sobolev gradient j' as the solution of the boundary value problem

$$\begin{cases} -\alpha_1 \mathrm{div}\left(\mathbf{grad}\left(\bar{j}' \right) \right) + \bar{j}' = -\omega^2 \sum_e u_e \overline{u}_e^\dagger & \text{in } \Omega, \\ \alpha_1 \dfrac{\partial \bar{j}'}{\partial n} = \dfrac{\partial \beta}{\partial s^2} \sum_e u_e \overline{u}_e^\dagger & \text{in } \partial\Omega. \end{cases} \tag{1.93}$$

4. Find the perturbed direct states δu_e obeying the perturbed direct problem (1.13)

$$\begin{cases} \mathrm{div}\left(\mathbf{grad}\left(\delta u_e \right) \right) + \omega^2 s^2 \delta u_e & = -\omega^2 \delta s^2 u_e & \text{in } \Omega \\ \dfrac{\partial \delta u_e}{\partial n} + \beta \delta u_e & = -\delta\beta u_e & \text{on } \partial\Omega. \end{cases} \tag{1.94}$$

5. Find the perturbed adjoint states $\delta u_e^\dagger$ obeying the perturbed adjoint problem

$$\begin{cases} \mathrm{div}\left(\mathbf{grad}\left(\delta \overline{u}_e^\dagger \right) \right) + \omega^2 s^2 \delta \overline{u}_e^\dagger & = \sum_r \delta \overline{u}_e(\boldsymbol{x}_r) \delta(\boldsymbol{x} - \boldsymbol{x}_r) - \omega^2 \delta s^2 \overline{u}_e^\dagger & \text{in } \Omega \\ \dfrac{\partial \delta \overline{u}_e^\dagger}{\partial n} + \beta \delta \overline{u}_e^\dagger & = -\delta\beta \overline{u}_e^\dagger & \text{on } \partial\Omega. \end{cases} \tag{1.95}$$

6. Once the perturbed direct and adjoint states are known, the second order directional derivatives with respect to the different model parameters can be evaluated, as well as the application of the Hessian operator in a particular direction:

   - second order directional derivative w.r.t. β, and associated application of the Hessian operator in the direction $\delta\beta$ by identification

$$\delta^2 J = \left\{ D^2_{\beta\beta} J(\beta) \right\} (\delta\beta_1, \delta\beta) = \sum_e \left\langle \delta u_e^\dagger, \delta\beta_1 \, u_e \right\rangle_{L_2(\partial\Omega)} + \left\langle u_e^\dagger, \delta\beta_1 \, \delta u_e \right\rangle_{L_2(\partial\Omega)} \quad (1.96)$$

hence

$$\delta\overline{j'} = \sum_e \delta_2 u_e \overline{u}_e^\dagger + u_e \delta_2 \overline{u}_e^\dagger. \quad (1.97)$$

- second order directional derivative w.r.t. s^2 and associated application of the Hessian operator in the direction δs^2 by identification

$$\delta^2 J = \left\{ D^2_{s^2 s^2} J(s^2) \right\} (\delta s_1^2, \delta s^2) = - \sum_e \left\langle \delta u_e^\dagger, \omega^2 \, \delta s^2 \, u_e \right\rangle_{L_2(\Omega)} + \left\langle u_e^\dagger, \omega^2 \, \delta s^2 \, \delta u_e \right\rangle_{L_2(\Omega)} \quad (1.98)$$

hence

$$\delta\overline{j'} = -\omega^2 \sum_e \delta u_e \overline{u}_e^\dagger + u_e \delta\overline{u}_e^\dagger \quad (1.99)$$

- second order directional derivative w.r.t. s^2, with β a function of s^2

$$\begin{aligned} \delta^2 J = \left\{ D^2_{s^2 s^2} J(s^2) \right\} (\delta s_1^2, \delta s^2) = & - \sum_e \left\langle \delta u_e^\dagger, \omega^2 \, \delta s^2 \, u_e \right\rangle_{L_2(\Omega)} + \left\langle u_e^\dagger, \omega^2 \, \delta s^2 \, \delta u_e \right\rangle_{L_2(\Omega)} \\ & + \sum_e \left\langle \delta u_e^\dagger, \frac{\partial \beta}{\partial s^2} \delta s^2 \, u_e \right\rangle_{L_2(\partial\Omega)} + \left\langle u_e^\dagger, \frac{\partial \beta}{\partial s^2} \delta s^2 \, \delta u_e \right\rangle_{L_2(\partial\Omega)} \end{aligned} \quad (1.100)$$

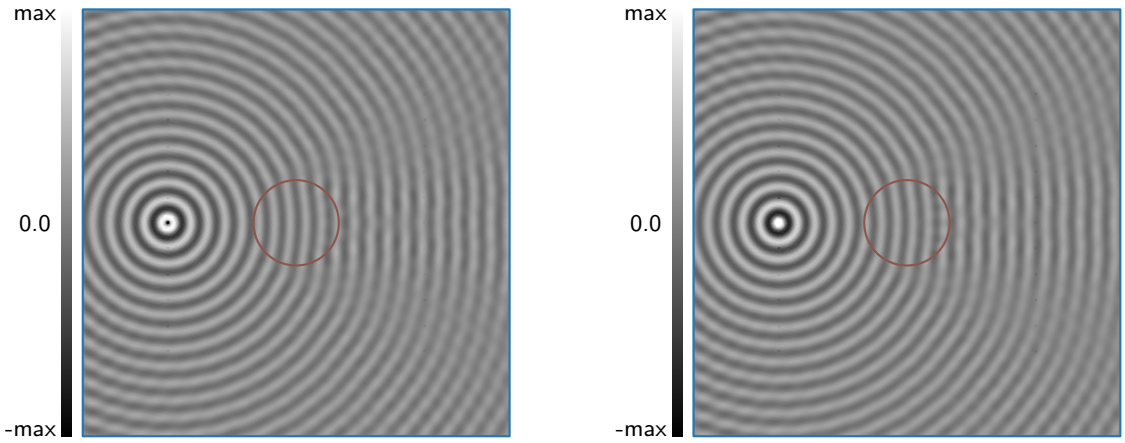
and the H_1-Sobolev Hessian operator in the direction δs^2 as the solution of the boundary value problem

$$\begin{cases} -\alpha_1 \mathrm{div}\left(\mathbf{grad}\left(\delta\overline{j'} \right) \right) + \delta\overline{j'} = -\omega^2 \sum_e \delta u_e \overline{u}_e^\dagger + u_e \delta\overline{u}_e^\dagger & \text{in } \Omega, \\ \alpha_1 \dfrac{\partial \delta\overline{j'}}{\partial n} = \dfrac{\partial^2 \beta}{\partial^2 s^2} \sum_e u_e \overline{u}_e^\dagger + \dfrac{\partial \beta}{\partial s^2} \sum_e \delta u_e \overline{u}_e^\dagger + u_e \delta\overline{u}_e^\dagger & \text{in } \partial\Omega. \end{cases} \quad (1.101)$$

For the sake of simplicity, the pulsation ω has been set to 2π, so that the wavelength is $\lambda = 1/s$. All partial differential equations are solved using the finite element method with 5^{th} order hierarchical elements, and a characteristic mesh size of $h = 1/4$. The measurements d are obtained with the same model, with the ground-truth values of the model parameters and a refined mesh of characteristic size $h = 1/5$.

1.6.1 Highly penetrable cylinder in a known background

The ground-truth solution of this reference problem is depicted in Figure 1.2 for a particular source, with $\tilde{s}_c^2 = 1.2$ and $\tilde{s}_0^2 = 1$ the ground-truth value of the slowness squared in the cylinder and the background medium, respectively.



(a) $\mathrm{Re}\,(u)$ (b) $\mathrm{Im}\,(u)$

Figure 1.2: Ground-truth direct field u for a highly penetrable cylinder and for a single source. The background and cylinder slowness squared are respectively $\tilde{s}_c^2 = 1.2$ and $\tilde{s}_0^2 = 1$.

In this first example, it is assumed that geometry and background medium are known, so that only the slowness squared of the cylinder is left unknown. Assuming it spatially uniform, one has $s_c^2 \in \mathbb{R}^+$ and

$$m(\boldsymbol{x}) := s^2(\boldsymbol{x}) = \begin{cases} \tilde{s}_0^2 & \text{for } \boldsymbol{x} \in \overline{\Omega}_0, \\ s_c^2 & \text{for } \boldsymbol{x} \in \Omega_c, \end{cases} \tag{1.102}$$

and thus

$$\delta m(\boldsymbol{x}) := \delta s(\boldsymbol{x}) = \begin{cases} 0 & \text{for } \boldsymbol{x} \in \overline{\Omega}_0, \\ \delta s_c^2 & \text{for } \boldsymbol{x} \in \Omega_c, \end{cases} \tag{1.103}$$

where $\overline{\Omega}_0$ denotes the closure of Ω_0, *i.e.* $\overline{\Omega}_0 := \Omega_0 \cup \partial\Omega_0$. It emphasizes the fact that both the bulk equation in Ω_0 and the transparent boundary condition on $\partial\Omega_0$ are depending on the model parameter s_0^2. In this first example however, the perturbation $\delta s^2(\boldsymbol{x})$ vanishes on the boundary $\partial\Omega_0$, and all boundary terms in (1.28) vanish, leaving only the bulk term. The performance functional and its Gâteaux derivatives reduce in this case to the real-valued functions $J(m(\boldsymbol{x})) = J(s_c^2)$, $\{D_m J(m(\boldsymbol{x}))\}(\delta m(\boldsymbol{x})) = D_{s_c^2} J(s_c^2)\delta s_c^2$ and $\{D_{mm}^2 J(m(\boldsymbol{x}))\}(\delta m_1(\boldsymbol{x}), \delta m_2(\boldsymbol{x})) = D_{s_c^2,s_c^2}^2 J(s_c^2)\delta s_c^2 \delta s_c^2$, which are plotted in Figure 1.3 for a range of values of s_c^2 around the ground-truth value $\tilde{s}_c^2 = 1.2$. For validation purposes, both derivatives are computed not only by the adjoint state method, but also by the finite difference approximation (1.3) with $\epsilon = 10^{-5}$. The two evaluations give results close to each other. The convergence of the finite difference approximation is further analyzed in Section 1.6.3.

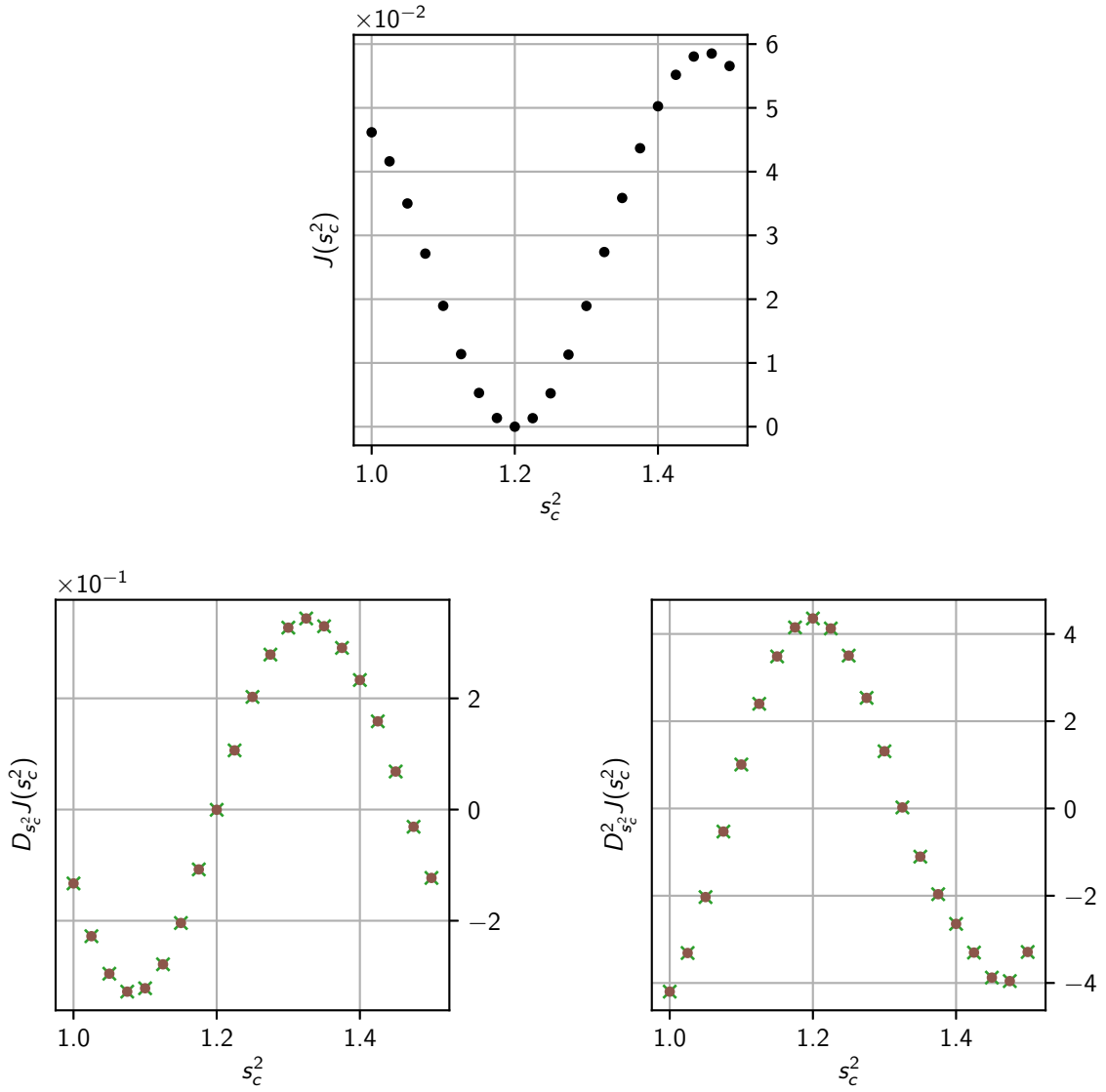


Figure 1.3: Performance functional (•) and its derivatives for a highly penetrable cylinder whose slowness squared s_c^2 varies around the ground-truth value $\tilde{s}_c^2 = 1.2$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

1.6.2 Weakly penetrable cylinder in a known background

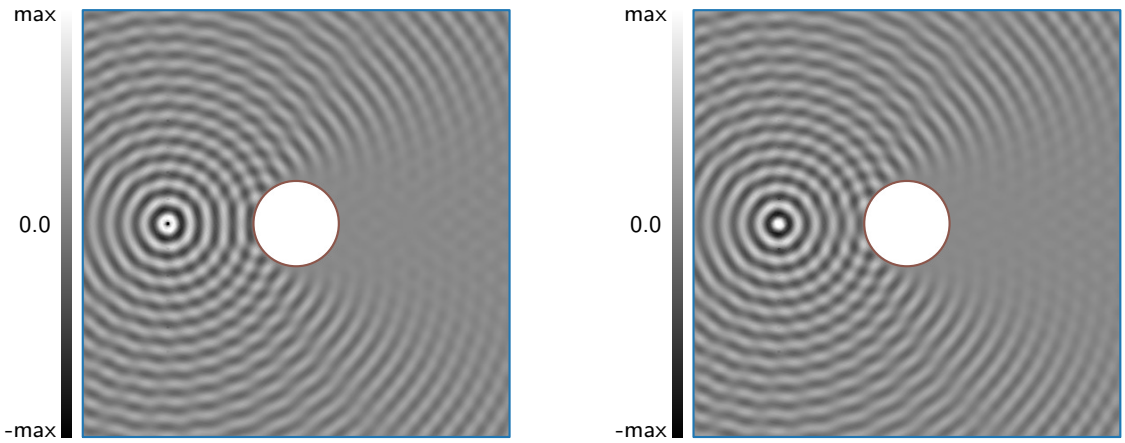
In this second example, the cylinder is considered weakly penetrable, and modeled by means of an equivalent impedance $\beta_c := i\omega\sqrt{\tilde{s}_c^2 - i\alpha_c}$ where $\alpha_c \in \mathbb{R}^+$ is a radially constant boundary parameter that dictates the penetrability of the cylinder. The model parameter writes

$$m(\boldsymbol{x}) := \alpha_c(\boldsymbol{x}) = \alpha_c \text{ for } \boldsymbol{x} \in \partial\Omega_c \tag{1.104}$$

and thus

$$\delta m(\boldsymbol{x}) := \delta\alpha_c(\boldsymbol{x}) = \delta\alpha_c \text{ for } \boldsymbol{x} \in \partial\Omega_c. \tag{1.105}$$

In contrast with the previous case, there is here no bulk contribution in (1.28), because α_c is a boundary parameter. The ground-truth direct field, computed with a ground-truth penetrability $\tilde{\alpha}_c = 100$, is depicted in Figure 1.4 for a particular source. The receiver array and the emitter array are placed on the same side of the cylinder in this case because, by an effect of shadowing, there is very little signal on the opposite side of the emitters. The performance functional and its Gâteaux derivative are plotted in Figure 1.5 for a range of values of α_c around the ground-truth value $\tilde{\alpha}_c = 100$.



(a) $\mathrm{Re}\,(u)$ (b) $\mathrm{Im}\,(u)$

Figure 1.4: Ground-truth direct field u for a weakly penetrable cylinder and for a single source. The background and cylinder slowness squared are respectively $\tilde{s}_c^2 = 1.2$ and $\tilde{s}_0^2 = 1$ while the penetrability is $\tilde{\alpha}_c = 100$.

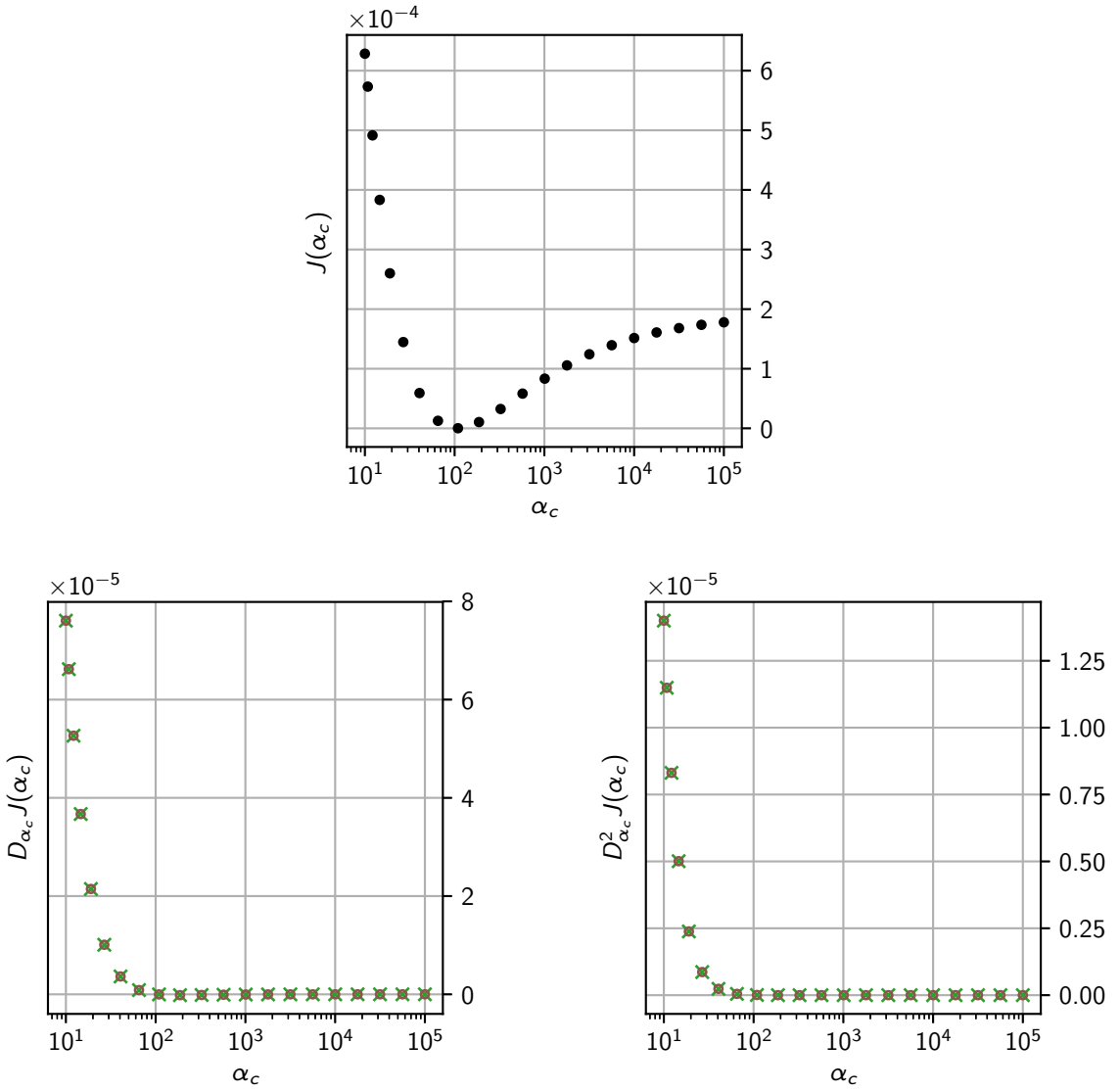


Figure 1.5: Performance functional (•) and its derivatives for a weakly penetrable cylinder whose penetrability α_c varies around the ground-truth value $\tilde{\alpha_c} = 100$. The derivatives are computed by the adjoint state method (○) and by an approximation of the definition (1.3)(with $\epsilon = 10^{-5}$) (×).

1.6.3 Background medium around a known highly penetrable cylinder

The case where the background slowness squared s_0^2 is unknown, whereas all properties of the cylinder are known, is now considered. One has

$$m(\boldsymbol{x}) := s^2(\boldsymbol{x}) = s_0^2 \text{ for } \boldsymbol{x} \in \overline{\Omega}_0 \setminus \Omega_c \tag{1.106}$$

and thus

$$\delta m(\boldsymbol{x}) := \delta s^2(\boldsymbol{x}) = \delta s_0^2 \text{ for } \boldsymbol{x} \in \overline{\Omega}_0 \setminus \Omega_c. \tag{1.107}$$

As the absorbing boundary condition $\beta_0 = i\omega\sqrt{s_0^2}$ on the outer boundary $\partial\Omega_0$ also depends on the slowness squared s_0^2 of the background domain, both the bulk and the boundary terms of the directional derivative (1.28) are present in this case. The performance functional and its Gâteaux derivatives are plotted in Figure 1.6 for a range of values of s_0^2 around the ground-truth value $\tilde{s}_0^2 = 1.0$, in the case of a highly ($\tilde{s}_c^2 = 1.2$, left row) and a weakly ($\tilde{\alpha}_c^2 = 100$, right row) penetrable cylinder.

Figure 1.6 shows that the boundary contribution is, for this example, much smaller than the bulk contribution. Neglecting it would however introduce a non-negligible error in the analytic evaluation of the derivative by means of the adjoint state method. To show this, the difference between the analytic and the finite difference evaluations is plotted in Figure 1.7 for decreasing values of ϵ, with and without the boundary contribution. This is done for a highly penetrable cylinder ($\tilde{s}_c^2 = 1.2$), and for a weakly penetrable cylinder ($\tilde{\alpha}_c = 100$) with a background slowness squared $s_0^2 = 0.93$, whereas the ground-truth value is $\tilde{s}_0^2 = 1.0$. As the finite difference converges towards the exact value of the derivative, up to numerical errors, as ϵ tends towards zero, it is indeed observed that the difference with analytic derivative decreases down to zero only when the boundary term is duly taken into account. For very small values of ϵ, the finite difference (1.3) becomes however sensitive to round-off errors, which explains the increasing tail in Figure 1.7.

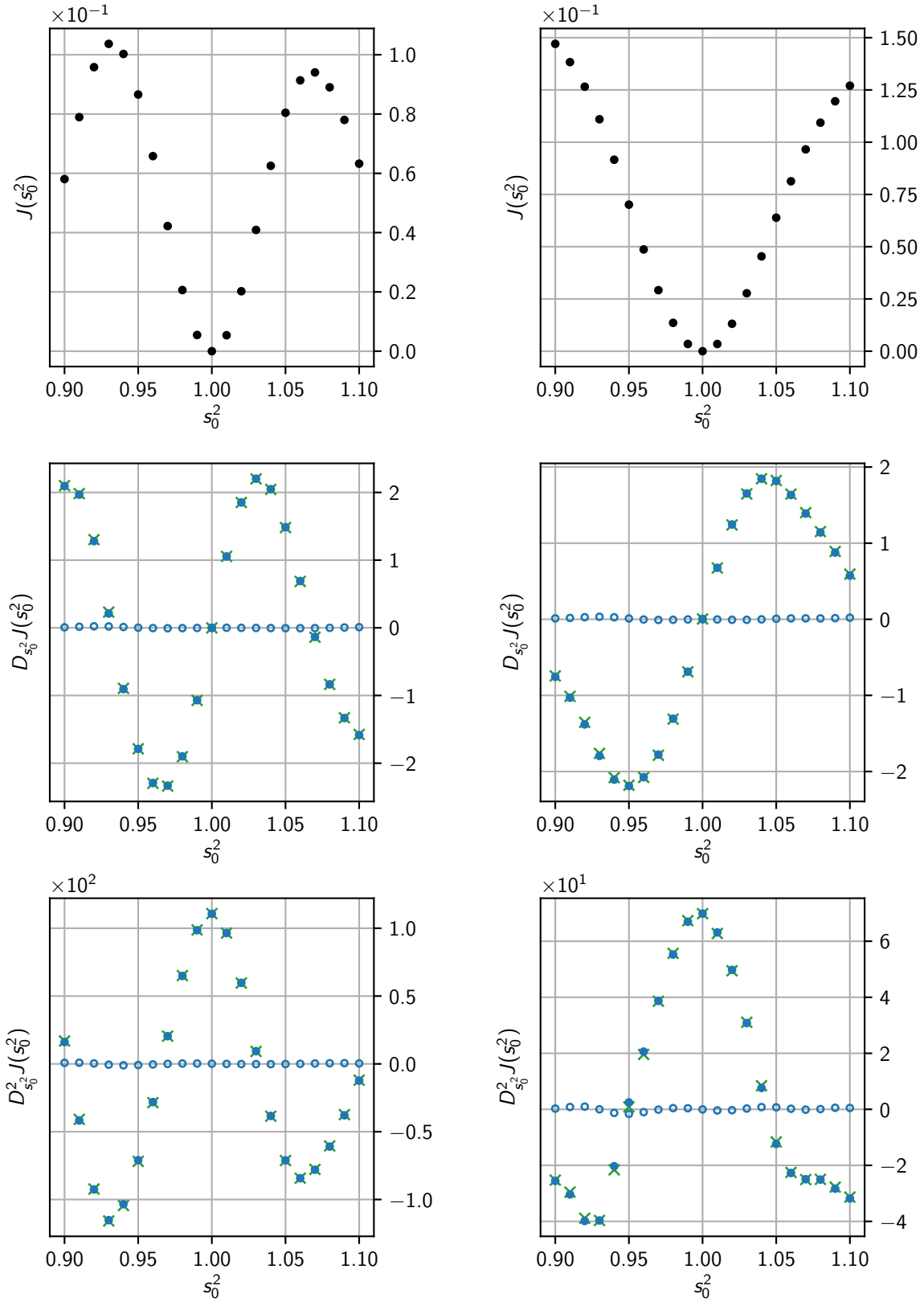


Figure 1.6: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{s}_c^2 = 1.2$)(left column) or a weakly penetrable cylinder ($\tilde{\alpha}_c = 100$)(right column) embedded in a background medium whose slowness squared s_0^2 varies around the ground-truth value $\tilde{s}_0^2 = 1.0$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

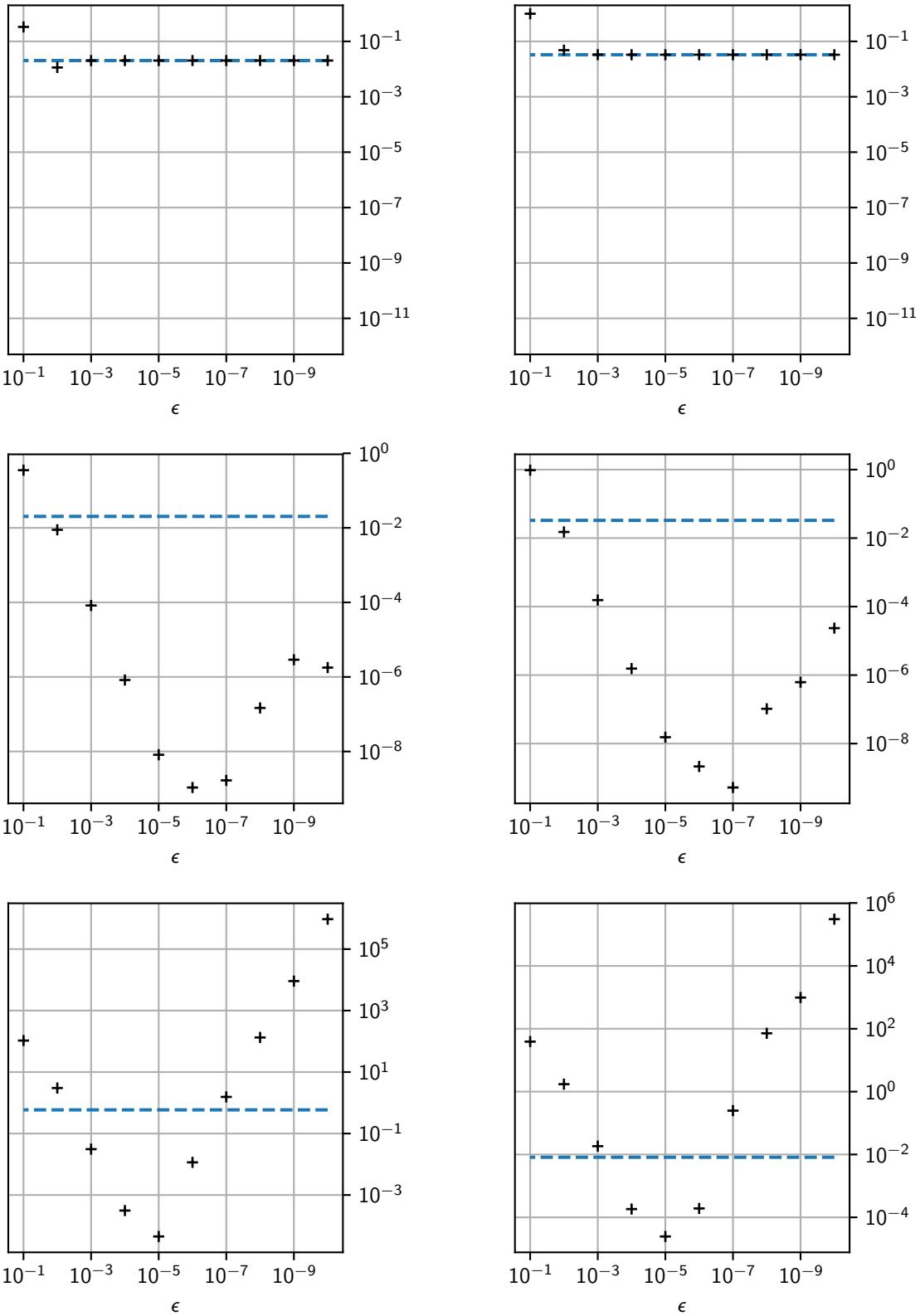


Figure 1.7: Difference between the approximation of (1.3) and the adjoint state method with (middle and bottom row) and without (top row) the boundary contribution for a highly penetrable cylinder ($\tilde{s}_c^2 = 1.2$)(left column) and for a weakly penetrable cylinder ($\tilde{\alpha}_c = 100$)(right column) when the unknown background slowness squared is $s_0^2 = 0.93$ while the ground-truth value is $\tilde{s}_0^2 = 1.0$. Bottom row is related to second order derivatives, middle and top row to first order derivatives. The dashed blue line (−−) is the amplitude of the boundary contribution in the adjoint state method.

1.6.4 Unknown geometry

The case is now considered where nothing is *a priori* known about the inclusion, not even its cylindrical shape. The measurement data are again obtained from the computation of a ground-truth problem with a highly penetrable cylinder ($\tilde{s}_0^2 = 1.0$ and $\tilde{s}_c^2 = 1.2$). The unknown squared slowness is, in this case, a distribution over Ω_0, and one is led by physical considerations to consider that this distribution is to be sought in a space of relatively smooth function, for instance $s^2 \in H_1(\Omega)$. Choosing then the natural inner product of $H_1(\Omega)$, the gradient j' is obtained as the solution of the boundary problem (*cf.* Subsection 1.5.3)

$$\begin{cases} -\alpha_1 \mathrm{div}\left(\mathbf{grad}\left(j'\right)\right) + j' = j'_\Omega & \text{in } \Omega_0, \\ \alpha_1 \dfrac{\partial j'}{\partial n} = j'_{\partial\Omega} \dfrac{\partial \beta_0}{\partial s^2} & \text{in } \partial\Omega_0. \end{cases} \tag{1.108}$$

The bulk and the boundary contributions to the gradient j' can be evaluated independently by setting successively to zero the boundary sensitivity $j'_{\partial\Omega}$ and the bulk sensitivity j'_Ω. Both contributions are depicted in Figure 1.8 for different values of the new parameter α_1, which defines the relative weight of the zeroth and the first order terms in the definition of the norm of $H_1(\Omega)$. The distinctive properties of the Sobolev gradient kernel j' are here illustrated by computing it at an initial guess, which is here naturally chosen as a homogeneous empty (*i.e.* without any inclusion) background, *i.e.*

$$m(\boldsymbol{x}) := s^2(\boldsymbol{x}) = \tilde{s}_0^2, \text{ for } \boldsymbol{x} \in \overline{\Omega}_0. \tag{1.109}$$

As in the previous example, the boundary contribution is much smaller than the bulk contribution. It is also more or less localized near the boundary $\partial\Omega_0$, depending on the value of α_1. The parameter α_1 thus controls how strongly the sensitivities j'_Ω and $j'_{\partial\Omega}$ are smoothed, as can be seen in Figure 1.8. This effect emphasizes the importance and the practical meaning of the choice of an appropriate inner product. An appropriate choice of the inner product is pivotal, as it may contribute to select gradients with interesting properties. In medium imaging in the time-harmonic regime, it is well-known that gradients are resolved at the wavelength scale. Consequently, the smoothing effect is apparent when the smoothing length defined as $l_c := 2\pi\sqrt{\alpha_1}$ is close to the wavelength $\lambda_0 = 1$. On the other hand, the gradient tends to become spatially uniform when the smoothing length is comparable to the size of the domain.

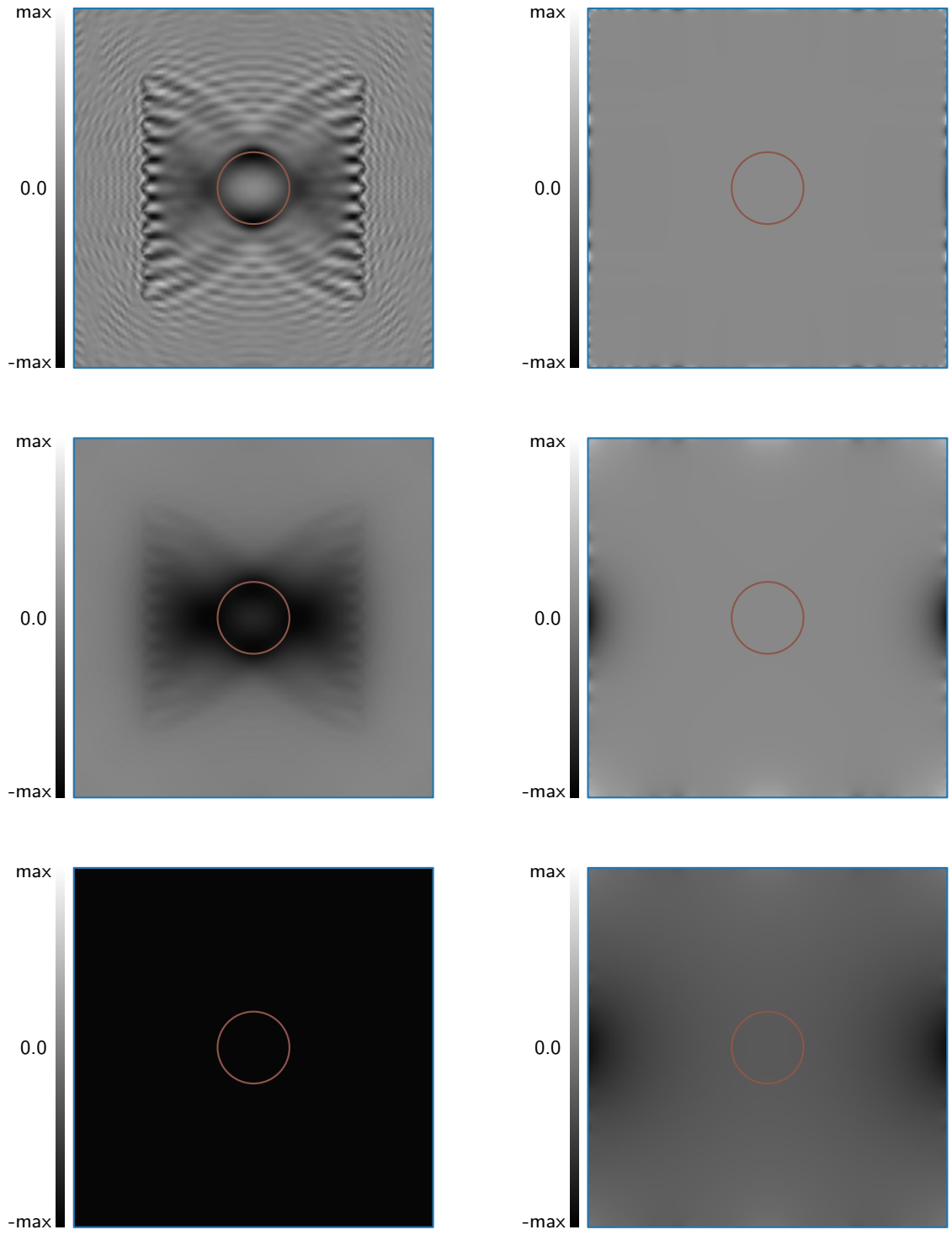


Figure 1.8: Bulk (left column) and boundary (right column) parts of the H_1-Sobolev gradient for several values of the smoothing length $l_c := 2\pi\sqrt{\alpha_1}$; $l_c = 1$ (top row), $l_c = 10$ (middle row), $l_c = 100$ (bottom row). The measured data used for this gradient are obtained synthetically with a highly penetrable cylinder ($\tilde{s}_0^2 = 1.0$ and $\tilde{s}_c^2 = 1.2$). From top to bottom, the maximal value for the bulk are boundary contribution are respectively 15×10^{-3}, 6×10^{-3}, 16×10^{-5} (bulk) and 3×10^{-3}, 2×10^{-4}, 7×10^{-6} (boundary).

1.7 Reference case studies

Since derivatives have been validated on a testing problem, the first step for the three reference case studies can now be carried out with confidence. This first step is the computation of the gradient direction, which is at the root of many local optimization algorithms. The wave propagation physics involved in these three reference imaging problems can also be modeled by the Helmholtz equation, complemented by Sommerfeld boundary conditions [163]. The first two case studies are related to subsurface acoustic imaging in the constant-density approximation, where the only model parameter to reconstruct is the pressure wave velocity v. In the context of the Helmholtz equation (1.83), the slowness squared, also called the sloth, is rather used as the model parameter. This sloth is simply the inverse of the squared velocity

$$s^2 := 1/v^2. \tag{1.110}$$

The entire derivation of previous section therefore remains valid: only the generic wave field u must be substituted by the pressure field p. The third numerical case study is related to electromagnetic imaging. When the electric field excitation is perpendicular to the direction of propagation, the electric field is also obtained by solving the Helmholtz equation. The generic wave field u must now be substituted by the transverse component e of the electric field and the wavenumber k ([1/m]) must be used in place of the pulsation ω ([rad/s]). Pulsation and wavenumber are related by the speed of light in vacuum, *i.e.* $k = \omega\sqrt{\epsilon_0\mu_0}$ with ϵ_0 the vacuum permittivity and μ_0 the vacuum permeability. The model parameter s^2 now denotes a dimensionless slowness squared, which is related to the conductivity and the relative permittivity through

$$s^2 := \epsilon_r - i\eta_0\sigma/k \tag{1.111}$$

where $\eta_0 := \sqrt{\mu_0/\epsilon_0} = 377$ [Ω] is the impedance of free space [97]. Sensitivities w.r.t. to the conductivity and the relative permittivity are then simply obtained from the sensitivities w.r.t. the dimensionless slowness squared using the chain rule.

Actually, only the performance functional is slightly modified w.r.t. the previous section. The conventional least squares distance is still chosen, but there is now also a summation on frequencies, in addition to the summation on emitters and receivers. Considering multiple frequencies introduces more information in the gradient and consequently, more constraints and resolution for the image to be reconstructed. For this chapter and the following one, recorded data sets are generated synthetically using the same hierarchical finite element settings than for the simulated data sets, in order to cancel the effect of numerical errors. This inverse crime approach will be dropped in Chapter 3.

1.7.1 Case study 1: Marmousi model

This first case study focuses on the Marmousi model, whose velocity distribution was given in the introduction (*cf.* Figure 3) and which is now plotted in terms of the slowness squared in Figure 1.9a.

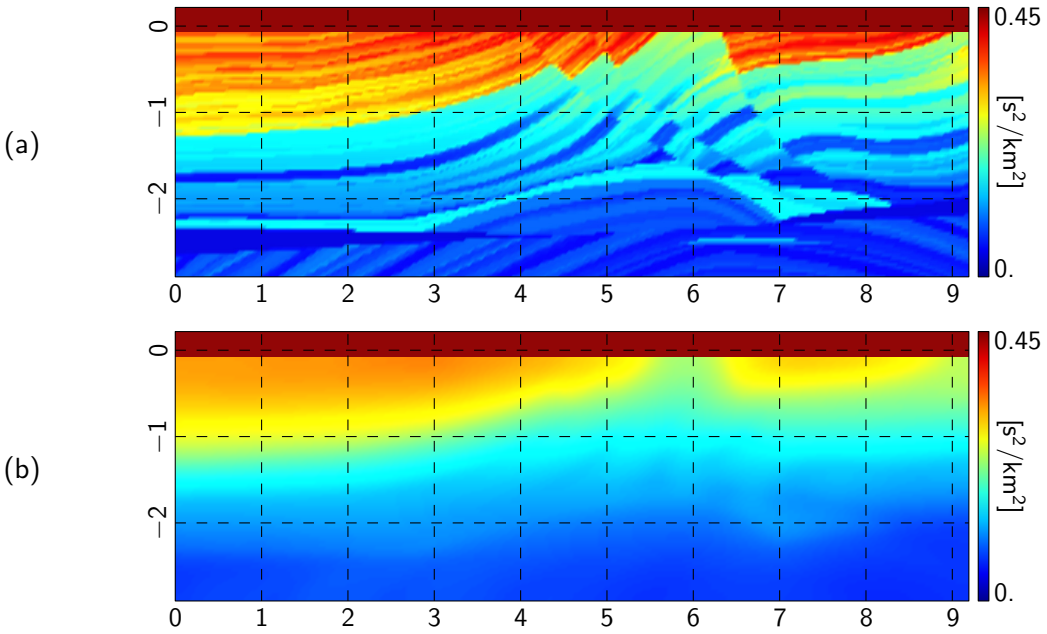


Figure 1.9: The Marmousi slowness squared model (a) and the chosen initial guess (b).

Three frequencies (4, 6 and 8 [Hz]) are included in the definition of the misfit. Frequencies lower than 4 [Hz] are not considered because they are typically not available in practice. The acquisition system is placed at the top surface of the model. It is composed of 122 equally spaced (72 [m]) excitation sources and 243 equally spaced (36 [m]) receivers. Slowness squared fields and pressure fields at the three frequencies are discretized on a square grid (36 [m]) by hierarchical finite elements, respectively of order 1 and of order 2, 3 and 4.

For the computation of any derivative, an initial guess is required for the model parameters. In this case study, a smoothed version of the exact Marmousi model is used as an initial guess (Figure 1.9b). This initial model is computed with a Laplacian filter $s^2_{\text{init}} = (1 - (l_c/2\pi)^2 \text{div}\,(\mathbf{grad}\,()))^{-1} s^2_{\text{exact}}$ with $l_c = 2$ [km]. In practice, such an initial guess could be obtained either thanks to travel-time tomography [206], through hierachical frequency-damping inversion strategies [22, 167], or with a data misfit functional which features less local minima [191, 115], for example. The H_1-Sobolev gradient, obtained with the smoothing inner product (1.79), is given in Figure 1.10 for a characteristic length $l_c = 2\pi\sqrt{\alpha} = 0.25$ [km]. Similarly to the cylinder testing problem, the boundary contribution has a much smaller contribution than the volume contribution, such that the former can be safely neglected for the sequel of this case study. This volume contribution however exhibits a striking weakness: it vanishes everywhere but at the top surface, where emitters and receivers are located. This

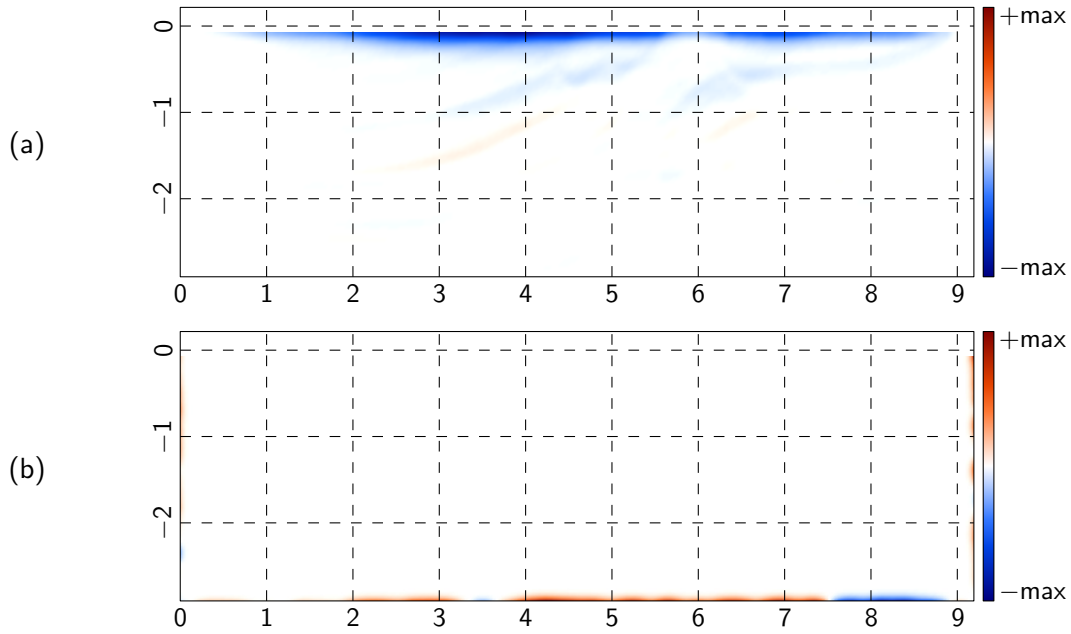


Figure 1.10: Bulk (a) and boundary (b) parts of the H_1-Sobolev gradient for a smoothing length $l_c = 2\pi\sqrt{\alpha_1} = 0.25$ [km]. The underlying slowness squared distribution is the initial guess given in Figure 1.9b. The maximal value for the bulk are boundary contribution are respectively 1020 and 22.

observation is the main motivation for the investigation of other inner products, which will specifically be studied and introduced in the next chapter.

1.7.2 Case study 2: T-shaped reflectors

The aim of this second case study is to image two T-shaped concrete structures ($v_c = 4$ [km/s]) embedded in a homogeneous background ($v_b = 0.3$ [km/s]) with a horizontal layer reflector in the bottom ($v_r = 0.5$ [km/s]). These two concrete foundations, buried at few meters deep, generate high-amplitude reflections because of the very high velocity contrast with the background. Moreover, important multiple scattering appears between the two structures, as they are relatively close to each other. The acquisition system is divided into three segments: one on the surface and the two others inside boreholes on both lateral sides. Sources and receivers are equally spaced (15 [cm]) along these three segments. Note that the surface sources and receivers that would lie inside the two concrete structures are not considered in the modeling, leading to an actual number of sources and receivers totaling 227. Nine frequencies (100, 125, 150, 175, 200, 225, 250, 275, and 300 [Hz]) are considered. Slowness squared fields and pressure fields at the nine frequencies are discretized on a square grid (15 [cm]) by hierarchical finite elements, respectively of order 1 and of order 2, 3 and 4. For the gradient evaluation, the initial model guess is simply the exact distribution without the two concrete structures, *i.e.* a homogeneous background with a bottom reflector. The H_1-Sobolev gradient is given in Figure 1.11 for two different characteristic lengths: $l_c = 0.5$

[m] and $l_c = 5$ [m]. Interestingly, the gradient, which is the foundation of the model update direction, exhibits ripples at the location of the two concrete structures when the smallest scale is used. These ripples are undesirable, as the velocity is constant inside the structures. They appear because a thin rippled interface and a piece-wise jump are hardly discernible at the considered frequencies. At the opposite, when the gradient is smoothed with more strength, ripples disappear and only the averaged contribution is conserved such that the gradient around the concrete structures is then in the right direction. Without this strong smoothing, the successive model updates might rush into a thin and oscillating direction and in the end, yield an inaccurate reconstruction which does not fit the *a priori* knowledge about the structures. This phenomenon will be studied in the next chapter.

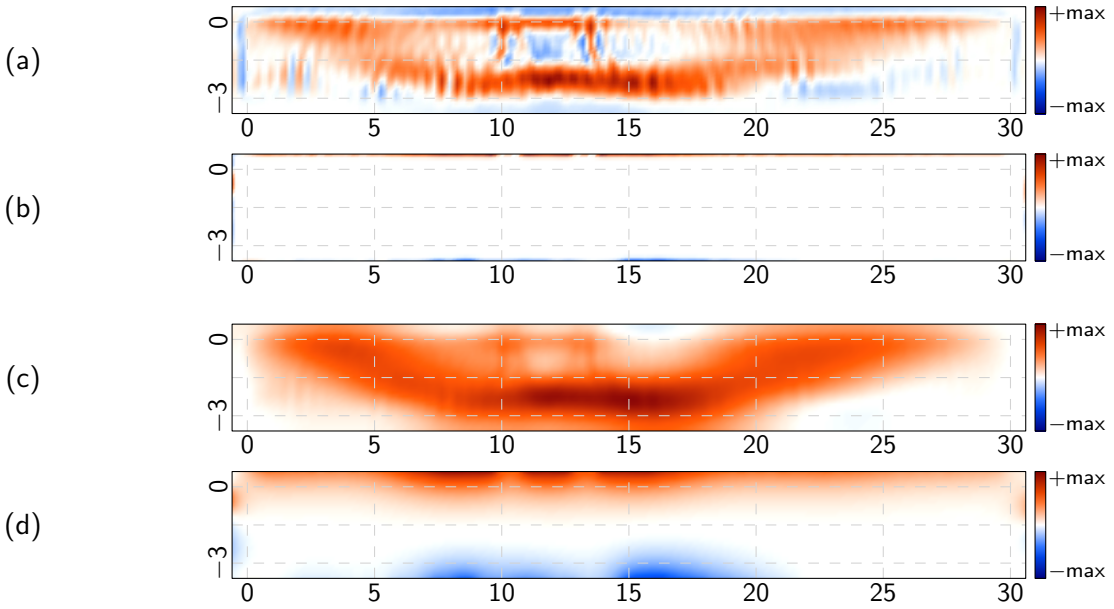


Figure 1.11: Bulk (a,c) and boundary (b,d) parts of the H_1-Sobolev gradient for two values of the smoothing length $l_c := 2\pi\sqrt{\alpha_1}$; $l_c = 0.5$ [m] (a,b), $l_c = 5$ [m](c,d). The underlying slowness squared distribution is a homogeneous background with a bottom reflector. From top to bottom, the maximal value for the bulk are boundary contribution are respectively 9×10^6, 3.2×10^6 (bulk) and 1.96×10^6, 1.7×10^5(boundary). Boundary contributions are again negligible.

1.7.3 Case study 3: Dissipative crosses

The third case study consists in two cross-shaped structures embedded in a homogeneous background (*cf.* Figure 5). The contrast between the background and the anomalies as well as the contrast between both anomalies is high ($\epsilon_{r,0} = 4.$, $\epsilon_{r,1} = 1.$, $\epsilon_{r,2} = 8.$ [-] and $\sigma_0 = 3.$, $\sigma_1 = 0.1$, $\sigma_2 = 10.$ [mS/m]) such that the two structures generate high-amplitude reflections. The acquisition system is composed of 120 equally spaced receivers placed on a circle ($r = 5$ [m]). Excitation sources are co-located with one of three receivers. There are thus only 40 excitation sources. Seven frequencies (50, 60, 70, 80, 100, 150, 200 [MHz])

are considered. Relative permittivity fields, conductivity fields and transverse electric fields at the seven frequencies are discretized on a triangular mesh (10 [cm]) by hierarchical finite elements, respectively of order 1 and of order 1, 2, 3 or 4. Similarly to the second case study, a homogeneous background is used as the initial guess. The conductivity and the relative permittivity gradients are given in Figure 1.11 for a characteristic length $l_c = 0.75$ [m]. With such an initial guess, the permittivity update, which is related to the displayed gradient, is going into the right direction: the value at the bottom cross will decrease while the value at the upper cross will increase. Unfortunately, the same can not be observed for the conductivity, as its value at the bottom cross will increase. Because the permittivity and thus the kinematics are not yet accurate, the conductivity is trying to take into account for effects caused by the permittivity and hence wrongly updates the conductivity. A strategy to mitigate this effect, which actually simply favors permittivty updates, will be proposed in the following chapter.

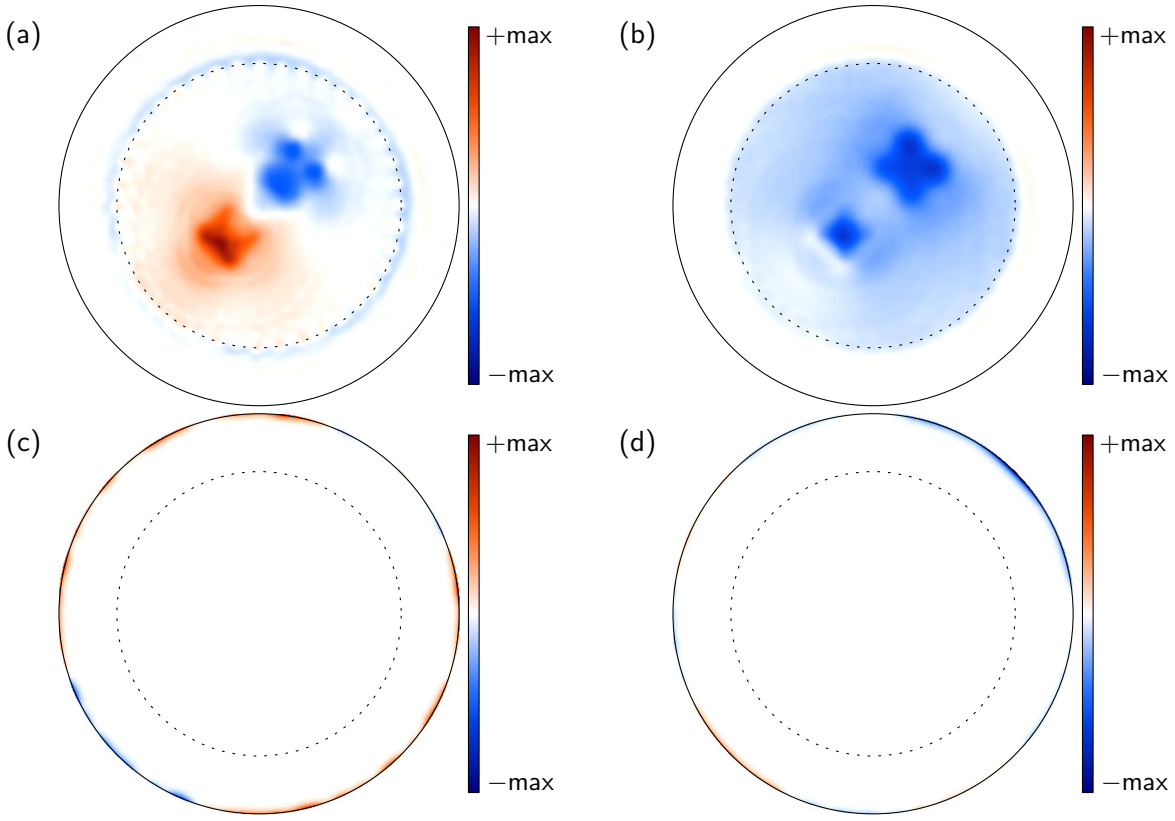


Figure 1.12: Bulk (a,b) and boundary (c,d) parts of the H_1-Sobolev gradient for the conductivity (b,d) or the relative permittivity (a,c) and for a smoothing length $l_c := 2\pi\sqrt{\alpha_1} = 0.75$ [m]. The underlying model parameters distribution is a homogeneous background. The maximal value for the bulk are boundary contribution are respectively 7.5×10^{-3}, 2.5×10^{-4} (relative permittivity) and 4.2×10^{-1}, 1.4×10^{-2} (conductivity). Boundary contributions are again negligible.

1.8 Conclusion

The adjoint state method is an elegant tool to compute derivatives efficiently when the performance functional depends on the model parameters through a state variable, being itself the solution of a partial differential problem. It is thus the first essential tool for full waveform inversion. First order derivatives can indeed be evaluated in that case independently of any model perturbations at the cost of solving one single extra linear system, while second order derivatives require solving only two more linear systems. In this chapter, the adjoint state method was extended to the problems where the model parameters modify not only the bulk properties of the computational domain but also the boundary conditions or boundary terms in the model. The procedure to evaluate gradient and Hessian kernels in the presence of boundary perturbations and using inner product preconditioning was also demonstrated.
Three generic time-harmonic wave scattering problems (acoustic, electromagnetic and elastodynamic waves) with model dependent boundary conditions have been treated in detail as application examples for the developed theory. Thanks to appropriate definitions, the adjoint state method was formulated in a unified way for the three problems, and for a large family of usual boundary conditions. Finally, the whole approach has been illustrated with a numerical testing problem, with boundary and bulk perturbations acting either independently or simultaneously. In particular, the computed derivatives were shown to be inexact when the boundary contribution is neglected and the strong influence of the inner product choice was highlighted. Finally, reference cases studies were investigated. Specifically, the first step towards the construction of an inversion procedure, *i.e.* the gradient computation, was performed and the need for preconditioning, for example through metric modifications, was highlighted. Now that the adjoint procedure to compute gradient and Hessian is set up, the next natural step is to combine these kernels to build optimization algorithms. This is the subject of the next chapter, whose goal is to use gradients, Hessian operators and inner product preconditioning in the more appropriate combinations to obtain the best inversion schemes in terms of convergence speed.

Chapter 2

Local optimization algorithms

Full waveform inversion is an imaging method which requires solving a large-scale minimization problem, typically through local optimization techniques. Most local optimization methods can basically be built up from two choices: the update direction and the strategy to control its length. In the context of full waveform inversion, this strategy is very often a line search and more rarely a trust-region method. Both are investigated in this chapter, in combination with non-standard inner products which act as preconditioners. More specifically, a line search and several trust-region variants of the steepest descent, the limited memory BFGS algorithm and the inexact Newton method are presented and compared. A strong emphasis is given to the inner product choice. For example, its link with preconditioning the update direction and its implication in the trust-region constraint are highlighted. Based on the three reference case studies, the importance of an appropriate inner product choice is highlighted and the best combination of methods is selected. In particular, it is illustrated that using an appropriate inner product greatly improves the convergence of all the presented methods and that inexact Newton methods should be combined with trust-region methods to increase their convergence speed.

Highlights

- Presentation and comparison of line search and trust-region globalization methods
- Illustration of the preconditioning properties of the inner product
- Comparison of the steepest, the l-BFGS and the Newton descent directions
- Assessment of their combinations based on the three reference case studies

2.1 Introduction

Full waveform inversion is formulated as a data fitting problem, whose aim is to recover some model parameters by minimizing the discrepancy between recorded data and data simulated by solving wave propagation problems [148, 189, 192]. By nature these data are oscillatory and consequently the misfit quantifying the discrepancy features local minima [19, 124]. Global optimization techniques should ideally be used but the typically very high dimensions of the search space prohibits their use and only local optimization tools can practically be employed, with care [36]. A straightforward direction to iteratively update the model properties is of course the gradient, *i.e.* the direction of steepest decrease. However, it is well-known that the inverse Hessian plays a crucial role in the reconstruction, in addition to offering the possibility to account for coupling effects between parameter classes for multi-parameter inversion [18, 116, 133, 134, 148, 201]. A theoretically simple way to incorporate these second-order derivatives is to minimize the misfit using Newton methods [2, 9, 18, 45, 113, 116, 134, 136, 135, 201]. In practice however, the pure Newton method is too computationally expensive to implement, because it requires inverting the Hessian operator. In addition, the misfit is not necessarily quadratic, thus the exact Newton direction is not necessarily appropriate. Consequently, it is natural to turn to inexact Newton methods, where the search direction is constructed iteratively to approximate the pure Newton direction, or to quasi-Newton methods. State-of-the-art methods rely on the quasi-Newton *l*-BFGS algorithm, which implicitly builds an approximation of the inverse Hessian operator from *l* previously saved gradients and model parameters [130]. However, it has been illustrated that on some specific cases involving multiple reflections, such quasi-Newton methods fail to converge where inexact Newton methods do succeed [118]. The latter compute the descent direction through a few iterations of a linear system involving the Hessian operator (the Newton system). One advantage over *l*-BFGS is the locality of the quadratic approximation: such methods do not rely on the convergence history of the algorithm, which might yield inaccurate inverse Hessian approximation for non quadratic misfit functions. The bottleneck of these methods lies in the compromise to find between a direction built in few iterations, but which hardly takes the Hessian into account and a nearly exact direction which is very expensive to compute. A complementary strategy to reduce this number of inner iteration is to apply a preconditioner to both sides of the Newton system [78, 26, 135, 196].

To implement any of the three above mentioned schemes, one can rely either on line search algorithms, or on trust-region methods. In the former case, once a direction is chosen, the outer iteration is completed by finding the optimal length of the step that should be performed along that direction. In the non linear optimization community, it is sometimes argued however that line search is not well suited with Newton directions, especially when the Hessian is nearly singular. Indeed when the Hessian is nearly singular, the Newton direction becomes excessively

long such that the local quadratic approximation implicitly made when computing it ceases to hold. Much computational effort must then be made by the line search procedure to reduce the step size [130]. Stopping the iterative solution of the Newton system earlier appears as a solution to this problem. For example, its convergence requirements could be relaxed such that they reflect the accuracy of the local quadratic approximation [43, 118]. Alternatively, a trust-region method could be used instead [102, 200, 205, 207]. The latter limits the length of the update direction depending on the accuracy of the local quadratic approximation. The length of a direction is given by its norm, itself induced by the inner product chosen for the model parameters space. The choice of this inner product is thus pivotal in the implementation of a trust-region method. Moreover changing the inner product modifies both the gradient and the Hessian and is equivalent to applying a preconditionner, as it was shown in Chapter 1. Consequently, it also has a major impact on line search based local optimization methods [34, 52, 127, 88, 210].
In this chapter, the three following important questions are tackled:

- Which descent direction to compute: the gradient, the l-BFGS direction or an inexact Newton direction?
- Which globalization method to select: a line search method or a trust-region method?
- Which preconditioning strategy to apply? How to enforce it?

Answering these three questions and determining the good combinations (good practices) between them is crucial for effective full waveform inversion. From this study, it appears that preconditioning is essential and that enforcing preconditioning through the inner product is elegant and, more interestingly, implies no modification to the practical implementation of the optimization algorithms. The *l*-BFGS method is found to be the most efficient method, in particular for both the single-parameter inversions. It is also found to be insensitive to the globalization choice. Inexact Newton methods should not be discarded though, as they perform nearly equally well for the double-parameter case study and more generally because considering the exact Hessian might lead to better model parameter decoupling in the case multi-parameter inversions. When using inexact Newton methods, the three reference case studies show that using a trust-region globalization consistently improves convergence.
The chapter is organized in two parts as follows. In the first part (Section 2.2), full waveform inversion is stated as a very general optimization problem. Its solution procedures using either a line search or a trust-region are introduced. The Newton system, which is pivotal in local minimization theory, is also re-derived. A particular emphasis is given to the inner product choice. More specifically, its link with preconditioning the Newton system is established. Local minimization methods commonly used in the context of full waveform inversion are then recalled. In the second part (Section 2.3), the application to imaging based on the Helmholtz

(I) equation is detailed. The computational cost of the adjoint procedure to compute gradients and Hessian vector products is also explained. The overall computational cost of each optimization method is then deduced. Finally, convergence results on the first case study, *i.e.* the Marmousi case study, are analyzed to determine the best parameters for a trust-region method. This best candidate is then compared to line search methods for the three reference case studies. The selection of the best inner product is also conducted, for each case study independently.

2.2 Local optimization methods

Full wave inversion is an imaging method based on the minimization of a misfit functional J, which exclusively depends on some model parameters m. The recovered model parameters m^* are defined as the minimizer of this misfit, *i.e.* $m^* = \arg\min J(m)$. Local optimization techniques are based on a local quadratic expansion of the misfit J around the current model estimate

$$J(m+\delta m) \approx J(m) + \{D_m J\}(\delta m) + \frac{1}{2}\{D^2_{mm} J\}(\delta m, \delta m). \tag{2.1}$$

This expansion can also be written in terms of the gradient j' and the Hessian operator H once an inner product $\langle \cdot, \cdot \rangle_M$ is chosen for the model space M

$$J(m+\delta m) \approx J(m) + \langle j', \delta m \rangle_M + \frac{1}{2} \langle H\delta m, \delta m \rangle_M . \tag{2.2}$$

The pure Newton direction p_N is then defined as the minimizer of this local quadratic expansion, which is also the solution of a linear system

$$p_N = \arg\min_{p \in M} J(m) + \langle j', p \rangle_M + \frac{1}{2} \langle Hp, p \rangle_M \quad \text{or} \quad Hp_N = -j'. \tag{2.3}$$

The large-scale nature of this linear system requires either the use of approximate Hessian operators that are straightforward to invert, or the use of Hessian-free iterative methods. These approaches are usually referred to as quasi-Newton methods and inexact Newton methods, respectively. In the latter case, the conjugate gradient method is the ideal candidate for the iterative solver because the Hessian operator is symmetric. The conjugate gradient method is however designed for positive definite operators while the full Hessian can be indefinite, especially far from the global minimum [118, 148]. As a consequence, either an additional safeguard is added to exit prematurely when directions of negative curvature are encountered or the exact Hessian is modified such that it becomes positive definite, *e.g.* using the Gauss-Newton approximation [135].

2.2.1 Globalization methods

As mentioned in the introduction, the misfit is not quadratic and thus the pure Newton direction or its approximations are not always the best directions. For that reason the length of the search direction is often tweaked using a line search or a trust-region method, which ensures convergence towards the nearest local minimum [43, 50, 51, 130].

Line search

When using a line search procedure, a direction p must first be identified. An appropriate length γ is then given to this direction p, ideally the global minimum along the line $m + \gamma p$. In practice however, less stringent satisfactory conditions are used instead to spare expensive wave problem resolutions. The most widely used example are strong Wolfe conditions

$$J(m + \gamma p) \leq J(m) + c_1 \gamma \{D_m J(m)\}(p) \tag{2.4}$$

$$|\{D_m J(m + \gamma p)\}(p)| \leq c_2 \, |\{D_m J(m)\}(p)| \tag{2.5}$$

for some constant c_1 and c_2 such that $0 < c_1 < c_2 < 1$. The first condition is called the *sufficient decrease condition*. It ensures that updating the model in the direction γp produces a decrease smaller than a fraction c_1 of what is expected from a local linear approximation of the misfit. The second condition, called the *curvature condition*, ensures that the updated model $m + \gamma p$ is sufficiently close to a local minimum along the line, where the directional derivative $\{D_m J(m + \gamma p)\}(p)$ would be zero. When this derivative is very smaller (resp. larger) than zero, then a larger (resp. smaller) step could produce a significantly bigger decrease. In this thesis, the chosen line search algorithm satisfies strong Wolfe conditions and accepts steps easily (Algorithm 3.2 from [130] with $c_1 = 10^{-4}$ and $c_2 = 0.9$). The outer loop is finally obtained by repeating these two steps iteratively until convergence.

Algorithm 1 Conventional line search algorithm

Require: m_0

loop

$$p_n = \begin{cases} -j'_n & \text{(steepest descent direction)} \\ \text{Algorithm 3} & \text{(}l\text{-BFGS direction)} \\ \text{Algorithm 5 with } \eta = \eta_n & \text{(inexact Newton direction)} \end{cases}$$

$$\gamma_n \approx \arg\min_{\gamma > 0} J(m_n + \gamma p_n)$$

$$m_{n+1} = m_n + \gamma_n p_n$$

end loop

Trust region

At the opposite when using a trust-region method, first a maximum length Δ is chosen. Then the best approximate solution, meaning the direction that minimizes a local prediction of the misfit but smaller than this length, is used

$$p = \underset{p\in M, \|p\|_M \leq \Delta}{\arg\min} \left[J^{\text{pred}}(m;p) := J(m) + \langle j'(m), p\rangle_M + 0.5 \left\langle \tilde{H}(m)p, p \right\rangle_M \right]. \tag{2.6}$$

This local misfit prediction J^{pred} is typically constructed based on the local quadratic approximation (2.2) through a particular choice of some approximate Hessian operator $\tilde{H}$. Of course the approximate Newton direction $\tilde{H}p = -j'$ is the solution of this problem if it lies inside the trust region. There are several possibilities to choose this length Δ and a particular choice is detailed later. More importantly, as pointed out in the introduction, the length constraint is formulated in terms of the norm induced by the inner product

$$\|p\|_M^2 = \langle p, p\rangle_M \leq \Delta^2. \tag{2.7}$$

Modifying this inner product therefore changes the shape of the trust region and it is then desirable to choose it carefully. The size of the trust region is actually controlled by the outer iterations. The decision of modifying the trust region is based on the accuracy of the local prediction of the misfit. When the prediction is accurate but the updates are limited by the length constraint, then the trust region radius is increased. Inversely, when the updates are out of the range of validity of the prediction, then the trust region radius is decreased. The decrease (resp. increase) rate of the radius is controlled by some parameter $c_0 < 1$ (resp. $c_1 > 1$). The quality of the prediction is quantified by the ratio between the actual decrease $\delta J_{\text{a}} := J(m_n) - J(m_{n+1})$ and the decrease predicted by the local prediction of the misfit.

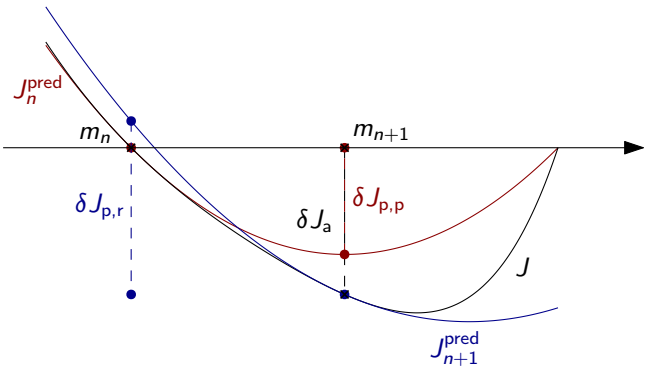


Figure 2.1: Illustration of the actual, prospective and retrospective predicted decreases for a mono-dimensional example where the predicted misfit is the exact second order expansion (Newton method).

There are two ways to compute this predicted decrease [51] (*cf.* Figure 2.1). On the one hand the expansion can be written in terms of the gradient and the Hessian operator at the

previous model estimate

$$J(m_{n+1}) = J(m_n + p_n) \tag{2.8}$$
$$\approx J(m_n) + \langle j'(m_n), p_n \rangle_M + 0.5 \left\langle \tilde{H}(m_n)p_n, p_n \right\rangle_M = J^{\text{pred}}(m_n; p_n) \tag{2.9}$$

which defines the *prospective predicted decrease*

$$\delta J_{\text{p,p}} := J(m_n) - J^{\text{pred}}(m_n; p_n) \tag{2.10}$$
$$= -\langle j'(m_n), p_n \rangle_M - 0.5 \left\langle \tilde{H}(m_n)p_n, p_n \right\rangle_M . \tag{2.11}$$

On the other hand, it can also be written in terms of the gradient and the Hessian operator at the next model estimate

$$J(m_n) = J(m_{n+1} - p_n) \tag{2.12}$$
$$\approx J(m_{n+1}) - \langle j'(m_{n+1}), p_n \rangle_M + 0.5 \left\langle \tilde{H}(m_{n+1})p_n, p_n \right\rangle_M = J^{\text{pred}}(m_{n+1}; -p_n) \tag{2.13}$$

which defines the *retrospective predicted decrease*

$$\delta J_{\text{p,r}} := J^{\text{pred}}(m_{n+1}; -p_n) - J(m_{n+1}) \tag{2.14}$$
$$= -\langle j'(m_{n+1}), p_n \rangle_M + 0.5 \left\langle \tilde{H}(m_{n+1})p_n, p_n \right\rangle_M . \tag{2.15}$$

These ratios between the actual decrease and one of both the predicted decreases

$$\rho_{\text{p}} := \frac{\delta J_{\text{a}}}{\delta J_{\text{p,p}}} \quad \text{and} \quad \rho_{\text{r}} := \frac{\delta J_{\text{a}}}{\delta J_{\text{p,r}}} \tag{2.16}$$

are actually both equal to one when the approximate Hessian in the update direction and the second order expansion (2.2) are exact. When the misfit is not quadratic or the Hessian approximation is not accurate, then these ratios can go away from one. Using anything else than the full Newton method can degrade these ratios, even if the misfit is quadratic. In particular for a pure quadratic misfit, neglecting the negative definite part of the Hessian makes the prospective ratio bigger than one ($\delta J_{\text{p,p}}$ is underestimated) and the retrospective ratio smaller than one ($\delta J_{\text{p,r}}$ is overestimated).

Standard trust-region methods directly control the radius Δ. However it is an absolute quantity, in the sense that it is compared to $\|p\|_M$, which depends on the inner product. Thus, it seems more natural to control this radius relatively to the gradient norm, *i.e.* $\Delta = \mu \|j'\|_M$, which provides a length reference for the (approximate) Newton system. In this way, even when the (approximate) Newton system changes scale from one iteration to another, the trust region remains relevant. This particular variant has been first introduced in [50] (*cf.* Algorithm 2).

Algorithm 2 Fan trust-region algorithm

Require: retrospective or prospective, $0 \leq \rho_0 < \rho_1 < 1$ and $0 < c_0 < 1 < c_1$

$\mu_0 = 1$

loop

$\Delta_n = \mu_n \|j'(m_n)\|_M$

$$p_n = \begin{cases} -\mu_n j'_n & \text{(steepest descent direction)} \\ (2.35) \text{ with } \Delta = \Delta_n & \text{(}l\text{-BFGS direction)} \\ \text{Algorithm 6 with } \Delta = \Delta_n & \text{(inexact Newton direction)} \end{cases}$$

$\delta J_{\mathrm{a}} = J(m_n) - J(m_n + p_n)$ and $\delta J_{\mathrm{p,p}} = J(m_n) - J^{\mathrm{pred}}(m_n; p_n)$

$\rho_{\mathrm{p}} = \delta J_{\mathrm{a}} / \delta J_{\mathrm{p,p}}$

if $\rho_{\mathrm{p}} \geq \rho_0$ **then** $m_{n+1} = m_n + p_n$ **else** $m_{n+1} = m_n$

if prospective **or** $\rho_{\mathrm{p}} < \rho_0$ **then**

$\rho = \rho_{\mathrm{p}}$

else if retrospective **then**

$\delta J_{\mathrm{p,r}} = J^{\mathrm{pred}}(m_{n+1}; -p_n) - J(m_{n+1})$

$\rho = \rho_{\mathrm{r}} = \delta J_{\mathrm{a}} / \delta J_{\mathrm{p,r}}$

end if

if $\rho < \rho_1$ **then** $\mu_{n+1} = c_0 \mu_n$

else if $\rho \geq \rho_1$ **and** $\|p_n\|_M > 0.5\Delta_n$ **then** $\mu_{n+1} = c_1 \mu_n$

else **then** $\mu_{n+1} = \mu_n$

end loop

According to this algorithm, a direction p_n is rejected when the prospective misfit prediction J_n^{pred} used to compute it is not accurate, in the sense that the prospective ratio is smaller than some threshold ρ_0. If not rejected, then the trust region size is updated according to either the prospective or the retrospective ratio, based on a comparison with a second threshold ρ_1. Because the updated radius Δ_{n+1} constrains the direction search around the next model estimate m_{n+1}, it makes sense to use the retrospective ratio which also involves the next model estimate m_{n+1} and not the prospective ratio which involves the current model estimate m_n. Using the retrospective ratio is however slightly more expensive because the next (approximate) Hessian operator in the current direction must be computed in addition. Moreover, the accuracy of the retrospective prediction might be good in the direction $-p_n$ while still being bad in the direction p_{n+1} and inversely. There are also no safeguards for large values of the ratios, which means that when the model is not accurate but the predicted decrease underestimates the true decrease, the radius can still be increased.

Three sets of values for the threshold ρ_1 and the rates c_0/c_1 have been tested (*cf.* Table 2.1). The acceptance threshold ρ_0 is always tiny such that steps are often accepted, similarly to the line search algorithm. The first one (A) is very similar to what was originally proposed in [51]. The other two (B,C) are more cautious because they modify the radius more rarely and when they do, it increases by a smaller factor. Note that the second one (B) is also close to what is proposed in [130].

	ρ_0	ρ_1	c_0	c_1
A	10^{-4}	0.25	0.20	5.
B	10^{-4}	0.75	0.25	2.
C	10^{-4}	0.90	0.50	2.

Table 2.1: Sets of values for the trust-region algorithm (Algorithm 2).

2.2.2 Inner product

The choice of the inner product plays a central role in the inversion because it defines, through the norm, how directions length are measured but also because it defines both gradients and Hessian operators. Indeed, it is interesting to remember that the equivalence between both expansions (2.1) and (2.2) is granted by the defining property of the gradient and the Hessian operator in terms of directional derivatives

$$\langle j', \delta m_1 \rangle_M := \{D_m J\}(\delta m_1) \qquad \forall \delta m_1, \tag{2.17}$$

$$\langle H\delta m_2, \delta m_1 \rangle_M := \left\{D^2_{mm} J\right\}(\delta m_1, \delta m_2) \quad \forall \delta m_1,\ \delta m_2. \tag{2.18}$$

The model parameter space is a function space defined on some region Ω and conventionally, the inner product is chosen as the $L_2(\Omega)$ inner product

$$\langle m_2, m_1 \rangle_M = \langle m_2, m_1 \rangle := \int_\Omega m_1(\boldsymbol{x}) \cdot m_2(\boldsymbol{x})\ d\Omega. \tag{2.19}$$

This straightforward choice leads to the conventional gradient j'_{L_2} and the conventional Hessian operator H_{L_2}, that can both be computed efficiently using the adjoint state method from Chapter 1. As an illustration, a conventional gradient is represented in Figure 2.2b. It is the first gradient computed during the acoustic imaging of the Marmousi model: nearly the same than in Chapter 2.2b actually. As can again be seen, shallow contributions have much greater amplitudes than deeper parts. This reflects the bad scaling properties of this inner product and motivates the use of a spatially weighted inner product

$$\langle m_2, m_1 \rangle_M := \left\langle m_2 \sqrt{w}, \sqrt{w}\ m_1 \right\rangle, \tag{2.20}$$

with an appropriate spatially dependant weight $w(\boldsymbol{x})$. Insights on how to design w can be gained by relating the conventional and the weighted gradients. Indeed, both are defined by (2.17) then by transitivity of the equality

$$\left\langle j'_{L_2}, \delta m_1 \right\rangle = \left\langle j' \sqrt{w}, \sqrt{w}\ \delta m_1 \right\rangle \quad \forall \delta m_1 \quad \text{such that} \quad j' = w^{-1} j'_{L_2}. \tag{2.21}$$

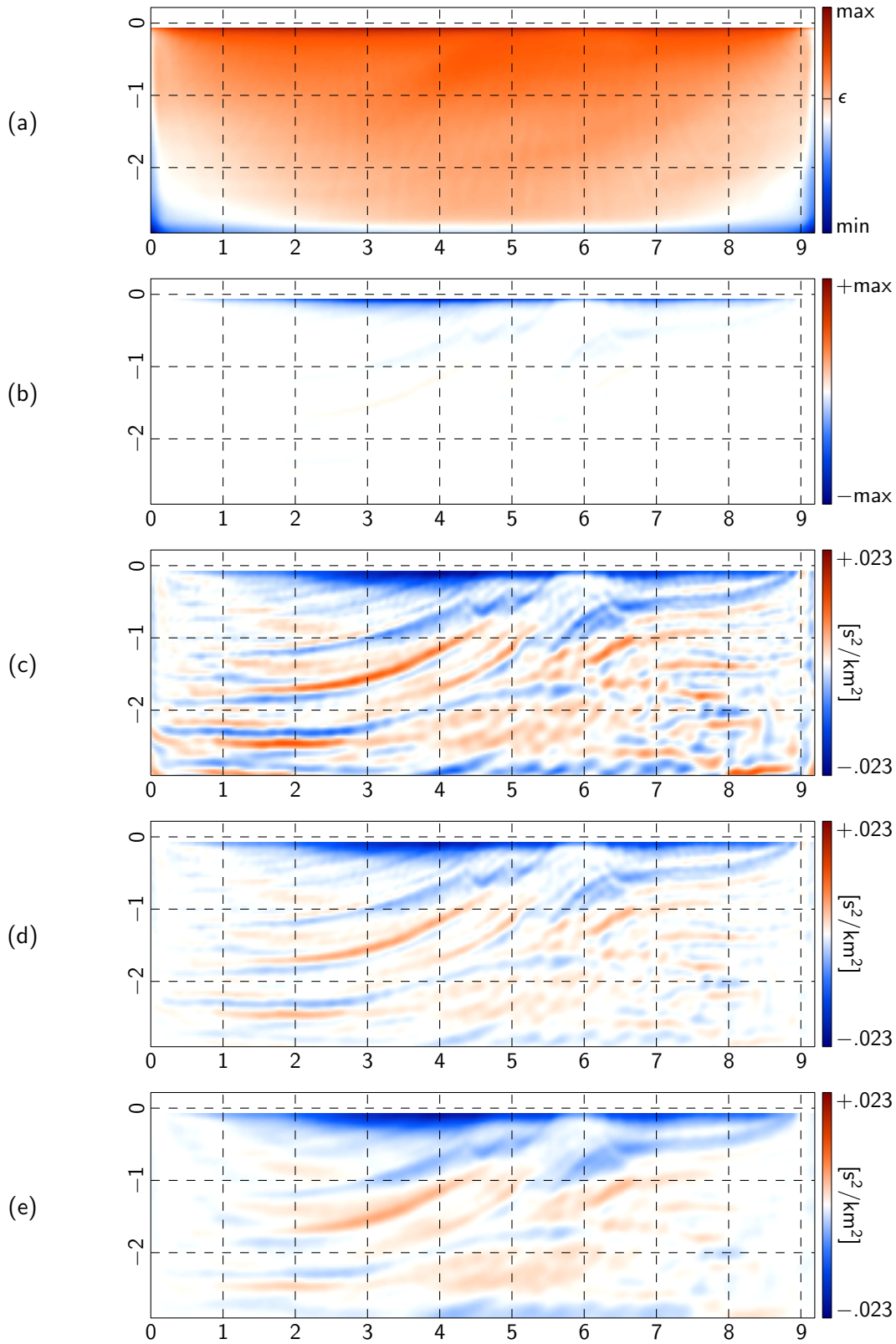


Figure 2.2: Diagonal part of the Gauss-Newton Hessian (a). Conventional gradient (b). Weighted gradient (c). Weighted and thresholded gradient (d). Weighted and smoothed gradient (e). The stabilization parameter ϵ is given graphically in (b). A smoothing length $l_c = 0.250$ [km]) is used for (e).

The same reasoning can be applied to both Hessian operator, *i.e.* $H = w^{-1} H_{L_2}$. Choosing this weight close to the Hessian operator then makes the gradient closer to the pure Newton direction and the Hessian operator closer to the identity. In other words, the Newton system (2.3) is better conditioned and iterative solvers are therefore expected to converge faster. This weight is here chosen as the diagonal part of the Gauss-Newton Hessian, *i.e.* $w = \mathrm{diag}\left(H_{\mathrm{GN}}\right)$, because it can be computed semi-analytically for a given model at no extra computational cost under certain circumstances (*cf.* Appendix C) [135]. A weight that has the same units than the Hessian also has the advantage that the corresponding weighted gradient has the same units than the model parameters. Model parameters, weighted gradients and weighted Hessian vector products therefore all have the units of model parameters and the coefficients between them, for example the length γ and μ involved respectively in line search and trust-region techniques, are then always dimensionless and thus easier to interpret. The weights and the corresponding weighted gradient are given in Figure 2.2a and 2.2c respectively. As expected, the weighted inner product compensates for the geometrical spreading and restores balance between shallow and deep contributions. It is however dangerous to use this weight alone because it can be very close to zero in poorly illuminated zones as for example in the corners of the model. In these regions, the weighted inner product is insensitive and consequently the preconditioner is unstable. The simplest stabilization strategy consists in the introduction of a threshold ϵ in the weights

$$\left\langle m_2, m_1 \right\rangle_M := \left\langle m_2 \ \sqrt{w}, \sqrt{w} \ m_1 \right\rangle + \epsilon \left\langle m_2, m_1 \right\rangle . \tag{2.22}$$

The corresponding preconditioning effect is to keep silent some regions, where the weight is much smaller than the threshold. Another strategy is to use an inner product with the following stabilization term

$$\left\langle m_2, m_1 \right\rangle_M := \left\langle m_2 \ \sqrt{w}, \sqrt{w} \ m_1 \right\rangle + \epsilon \, \alpha \left\langle \mathbf{grad}\left(m_2\right), \mathbf{grad}\left(m_1\right) \right\rangle \tag{2.23}$$

where $\alpha := \left(l_c / 2\pi\right)^2$ and l_c is characteristic length. This second term, related to spatial derivation, increases the norm of directions that are rapidly varying and also prevents the inner product from being insensitive in regions where the diagonal Hessian is close to zero. In regions where the diagonal Hessian is close to the threshold, then directions with details smaller than the characteristic length l_c are penalized with respect to smoother directions. This inner product is actually very similar to the one introduced in [210], except that the Gauss-Newton diagonal Hessian weight is used in addition. As far as preconditioning is concerned, this inner product can be reformulated through an integration by parts as

$$\left\langle m_2, m_1 \right\rangle_M := \left\langle w \ m_2, m_1 \right\rangle - \epsilon \, \alpha \left\langle \mathrm{div}\left(\mathbf{grad}\left(m_2\right)\right), m_1 \right\rangle . \tag{2.24}$$

Then as in Chapter 1, conventional and preconditioned gradients are linked

$$\langle j', \delta m_1 \rangle_M = \left\langle j'_{L_2}, \delta m_1 \right\rangle \quad \forall \delta m_1 \tag{2.25}$$

$$\langle (w - \epsilon\, \alpha \mathrm{div}\,(\mathbf{grad}\,()))\, j', \delta m_1 \rangle = \left\langle j'_{L_2}, \delta m_1 \right\rangle \quad \forall \delta m_1 \tag{2.26}$$

$$\Leftrightarrow \quad j' = (w - \epsilon\, \alpha \mathrm{div}\,(\mathbf{grad}\,()))^{-1}\, j'_{L_2}. \tag{2.27}$$

From the point of view of preconditioning, this inner product generates a rescaling thanks to the Gauss-Newton diagonal Hessian weight and a Laplacian filtering, whose smoothing length equals l_c in regions where the diagonal Hessian equals the threshold, *i.e.* where $w(\boldsymbol{x}) \approx \epsilon$. In regions where this weight is smaller (resp. larger) than the threshold then the smoothing is stronger (resp. weaker). The effect of these inner products is illustrated in Figure 2.2d and 2.2e. In addition to stabilizing the weights, filtering inner products have been shown to help the convergence of full waveform inversion by mitigating its non linearity[210].
In general, any inner product that can be related to the conventional inner product (2.19) through some preconditioner P, yields a preconditioned gradient and a preconditioned Hessian operator

$$\langle m_2, m_1 \rangle_M = \langle P m_2, m_1 \rangle \quad \Rightarrow \quad j' = P^{-1} j'_{L_2} \text{ and } H = P^{-1} H_{L_2}. \tag{2.28}$$

Changing the inner product is formally equivalent to preconditioning both the gradient and the Hessian operator. It is here chosen to introduce preconditioning through a change in the inner product rather than through the application of an operator because it appears more elegant and rigorous. Moreover, this approach has the pedagogical advantage to include preconditioning inside the inner product choice and thus it does not need to appear explicitly in the description of the optimization algorithms. In terms of practical implementation, it implies that the optimization routines need not be rewritten, only the subroutine which computes the inner product has to be modified, hence providing a lot of flexibility. Basically, a different choice for the inner product does not modify the pure Newton direction because the same preconditioner is applied to both sides of the Newton system (2.3), but does modify the subspace constructed by the conjugate gradient method and does modify norms which are involved in any stopping criterion. A good choice can thus lead to better approximate directions and better truncation rules.

2.2.3 Steepest descent

The steepest descent is actually the simplest local gradient-based optimization algorithm. It consists in taking the search direction as the opposite gradient. This is the best direction at first order ($\tilde{H} = 0$) but it can also be seen as a quasi-Newton step where the approximate Hessian operator is the identity operator ($\tilde{H} = I$). In practice however, this approximation is very crude because the Hessian operator is far from the identity operator, even after precondition-

ing. The downside of this simple method is its linear convergence rate. This slow convergence speed is one of the main motivation for the investigation of higher order algorithms.

Line search globalization

No length information can be captured from the approximate Hessian operator in this case, because it is simply the identity operator ($\tilde{H} = I$). The first trial step length is then chosen based on the history of the outer iterations to save as many step length trials as possible [130], *e.g.*

$$\gamma = 2(J(m_n) - J(m_{n-1}))/\{D_m J\}(-j'). \tag{2.29}$$

Trust region globalization

Trust-region methods are barely used with steepest descent. Mostly because the linear misfit prediction

$$J^{\text{pred}}(m; p) := J(m) + \langle j'(m), p\rangle_M \tag{2.30}$$

is not accurate enough. Moreover the solution to the trust-region sub-problem (2.6) is trivially $p = -\mu j'$ and is always on the boundary, because of the absence of a second order term. An upper bound on the relative size of the trust region μ is then added to compensate the fact that the trust-region algorithm will never keep it constant. This bound is set to $\mu_{\max} = 4$, 4, 5 for parameter sets A, B, C respectively.

2.2.4 Limited memory BFGS method

Quasi-Newton methods are expected to provide a huge improvement over the steepest descent and an attractive alternative to Newton methods because they do not involve any expensive Hessian vector product. In place of the exact Hessian, an approximation $\tilde{H} = B$ is used instead. This approximation is built only with the successive gradients and model parameters of each iteration. Moreover, since expensive Hessian vector product are avoided, quasi-Newton methods are sometimes more efficient than Newton methods. The Broyden-Fletcher-Goldfarb-Shanno algorithm, abbreviated BFGS, is maybe the most widely used quasi-Newton method. This method constructs a symmetric and positive definite approximation of the Hessian operator based on all the previous gradients and model parameters. This approximation B_{n+1} is chosen such that it verifies the secant equation

$$B_{n+1}\Delta m_n = \Delta j'_n \quad \text{with } \Delta m_n = m_{n+1} - m_n \text{ and } \Delta j'_n = j'_{n+1} - j'_n \tag{2.31}$$

while being close to the previous approximation B_n and positive definite. Note that imposing the positive definiteness of this approximation also imposes that the update direction must satisfy the (BFGS) curvature condition $\langle \Delta m_n, \Delta j'_n\rangle_M > 0$. One of the biggest advantage of

the BFGS algorithm is that it is possible to directly build the approximate inverse Hessian operator B_n^{-1} from the memorized gradients and model parameters. However, building explicitly this inverse operator in the context of large-scale optimization is still prohibitively expensive, as well as storing in memory all the previous gradients and model parameters. For these reasons, a limited memory version of the algorithm has been derived. Instead of memorizing all the previous iterates, it only requires the l last iterates and above all, it comes with a two-loop recursive procedure to compute the application of the inverse operator on any direction. The approximate Newton direction associated with the l-BFGS operator is therefore straightforward to compute. This two-loop recursive l-BFGS algorithm is given in Algorithm 3. A constant initial Hessian approximation, *i.e.* $B_n^0 = \left\langle \Delta m_{n-1}, \Delta j'_{n-1} \right\rangle_M / \left\langle \Delta j'_{n-1}, \Delta j'_{n-1} \right\rangle_M$, is here chosen [130].

Algorithm 3 Inverse l-BFGS operator application

Require: q, Δm_k, $\Delta j'_k$, $\quad \forall k \in [n-l, n-1]$

for $k = n-1$ **down to** $k = n-l$ **do**
  $\alpha_k = \langle \Delta m_k, q \rangle_M / \langle \Delta j'_k, \Delta m_k \rangle_M$
  $q = q - \alpha_k \Delta j'_k$
end for

$r = B_n^0 q$

for $k = n-l$ **up to** $k = n-1$ **do**
  $\beta_k = \langle \Delta j'_k, r \rangle_M / \langle \Delta j'_k, \Delta m_k \rangle_M$
  $r = r + (\alpha_k - \beta_k) \Delta m_k$
end for

return $r \ (= B_n^{-1} q)$

It is important to highlight here that this method also benefits from the modification of the inner product. Indeed, the building blocks of this approximate Hessian operator are the successive gradients, which are preconditioned through the inner product. By measuring gradient variations, this method constructs a representation of the misfit which is good enough to produce super-linear convergence, a great improvement over the steepest descent, at no extra cost. This approximation is however positive definite while the exact Hessian might be indefinite, especially during the early iterations of the inversion. In such cases, this quasi-Newton method may fail to converge while Newton methods may not [118].

Line search globalization

The unit step length $\gamma = 1$ is always tried first because the length information should be captured by the inverse approximate Hessian. Importantly, it can be shown that the (BFGS) curvature condition is always satisfied if the strong Wolfe conditions (2.4) and (2.5) are

enforced [130]. Therefore the *l*-BFGS algorithm combined with a line search will always construct a positive definite approximate Hessian operator B.

Trust region globalization

Finding the exact solution to the trust-region sub-problem (2.6) with the *l*-BFGS predicted misfit

$$J^{\text{pred}}(m;p) := J(m) + \langle j'(m), p\rangle_M + 0.5\,\langle Bp, p\rangle_M \tag{2.32}$$

is difficult for a general trust region radius. However when this radius is large enough, in particular larger than the unconstrained solution $p^{\text{u}} := -B^{-1}j'$, then it is actually also the exact constrained solution. On the other hand, when the radius is small enough, the quadratic term in the misfit prediction is negligible and the sub-problem is equivalent to the steepest descent, which indicates following the gradient up to the boundary. Based on these solutions for the extreme value of the radius, the exact solution to the sub-problem (2.6) might be substituted by an interpolation between these two solutions.

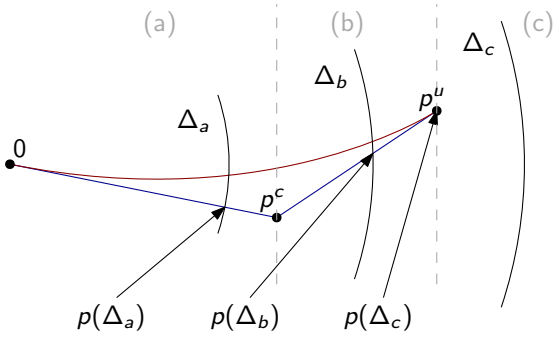


Figure 2.3: Illustration of the dogleg method (dogleg (▬), exact (▬)).

Namely, the gradient is followed each time the minimum of the misfit prediction along the gradient, *i.e.* the Cauchy point $p^{\text{c}} = -\alpha j'$ (with $\alpha = \langle j', j'\rangle_M / \langle Bj', j'\rangle_M$), is outside the radius

$$p(\Delta) = -\frac{\Delta}{\|j'\|_M} j' = -\mu j' \quad \text{when} \quad \mu \leq \alpha. \tag{2.33}$$

Then for intermediate radii, which contains this Cauchy point but not the unconstrained solution, an interpolation between both is done

$$p(\Delta) = p^{\text{c}} + \tau^*\,(p^{\text{u}} - p^{\text{c}}) \;\text{ with } 0 < \tau^* < 1 \text{ such that } \;\|p\|_M = \Delta. \tag{2.34}$$

Finally for large radii, the unconstrained solution is accepted. In summary

$$p(\Delta) = \begin{cases} p^{\text{u}} & \text{when } \|p^{\text{u}}\|_M \leq \Delta \quad \text{(c)}, \\ -\mu j' & \text{when } \|p^{\text{c}}\|_M \geq \Delta \quad \text{(a)}, \\ p^{\text{c}} + \tau^*(p^{\text{u}} - p^{\text{c}}) & \text{when } \|p^{\text{c}}\|_M \leq \Delta \leq \|p^{\text{u}}\|_M \quad \text{(b)}. \end{cases} \tag{2.35}$$

The approximate solution (2.35) to the trust-region sub-problem (2.6) is called the *dogleg method* [130]. It is illustrated schematically in Figure 2.3.
A huge difference with the line search implementation of the *l*-BFGS algorithm is that now the direct application of the approximate Hessian operator B on some directions must be computed. Unfortunately there is no equivalent to Algorithm 3 for the direct *l*-BFGS operator and its application must then be computed from its recursive definition at iteration n, *i.e.*

$$B_i q = B_n^0 q + \sum_{k=n-l}^{i-1} b_k \langle b_k, q \rangle_M - a_k \langle a_k, q \rangle_M$$
$$\text{with} \quad a_k = \frac{B_k \Delta m_k}{\sqrt{\langle B_k \Delta m_k, \Delta m_k \rangle_M}} \quad \text{and} \quad b_k = \frac{\Delta j'_k}{\sqrt{\langle \Delta j'_k, \Delta s_k \rangle_M}}. \tag{2.36}$$

It is important to highlight that the sequence of directions a_k could not be memorized because at each iteration the oldest information is discarded, which modifies the whole a_k sequence. A complete procedure to compute the application of the direct *l*-BFGS operator is given in Algorithm 4. Faster but more sophisticated procedures do exist [130]. However, manipulations in the model parameter space are computationally negligible with respect to wave propagation problems, hence the speedup would also be negligible. Thanks to this procedure the prospective and retrospective predicted decrease (2.11) and (2.15) can be evaluated. Interestingly, the prospective decrease is evaluated with the current Hessian approximation B_n while the retrospective decrease is evaluated with the next Hessian approximation B_{n+1}. The retrospective ratio is therefore expected to be more often close to one because this next Hessian approximation B_{n+1} is specifically constructed from the update direction $p_n = \Delta m_n = m_{n+1} - m_n$.

Algorithm 4 Direct *l*-BFGS operator application

Require: $q,\ \Delta m_k,\ \Delta j'_k, \quad \forall k \in [n-l, n-1]$

for $k = n - l$ **up to** $k = n - 1$ **do**
  $b_k = \Delta j'_k / \sqrt{\langle \Delta j'_k, \Delta m_k \rangle_M}$
  $a_k = B_n^0 \Delta m_k$
  for $i = n - l$ **up to** $i = k - 1$ **do**
    $a_k = a_k + \langle b_i, \Delta m_k \rangle\, b_i - \langle a_i, \Delta m_k \rangle\, a_i$
  end for
  $a_k = a_k / \sqrt{\langle \Delta m_k, a_k \rangle_M}$
end for

$r = B_n^0\, q$
for $k = n - l$ **up to** $k = n - 1$ **do**
  $r = r + b_k \langle b_k, q \rangle_M - a_k \langle a_k, q \rangle_M$
end for

return $r\ (= B_n\, q)$

2.2.5 Newton methods

In contrast to quasi-Newton methods, Newton methods use the Hessian operator explicitly, as they try to solve the Newton system (2.3). The interest of these methods lies in their independence on the convergence history and in their quadratic convergence rate in the vicinity of the minimum. Far from this minimum, the Hessian operator might however be indefinite, which complicates the solution procedure for the Newton system. For that reason, it is frequent to make the Gauss-Newton approximation ($\tilde{H} = H_{\mathrm{GN}}$), which consists in keeping only the positive definite part of the Hessian operator (*cf.* Appendix C). The downside of this approximation is then that the second order representation (2.2) of the misfit is less accurate, especially if the negative definite part of the Hessian is not negligible, which might prevent the method from converging. In this section, inexact Newton methods based on a line search procedure or a trust-region method are introduced. Both are valid for the full Hessian and for its Gauss-Newton approximation.

Line search globalization

Newton methods can be combined with a line search procedure. In this case a direction p is first found by solving the Newton system approximately with the conventional conjugate gradient method (*cf.* Algorithm 5) [130]. This algorithm constructs iteratively the solution of a linear system without requiring the explicit expression of the Hessian matrix but only its action in particular directions. The iterative procedure is stopped when the residuals have decreased more than some threshold, called the *forcing sequence* η, which is typically close to zero, *i.e.*

$$(\|r_k\|_M :=) \, \|Hp_k + j'\|_M < \eta \, \|j'\|_M \, (= \eta \, \|r_0\|_M) \, . \tag{2.37}$$

Over-solving is here avoided through this forcing term η, which is not systematically close to zero but which is instead chosen to reflect the accuracy of the second-order expansion.

Algorithm 5 Conventional conjugate gradient algorithm

Require: $0 < \eta \leq 1$

if $\langle Hj', j' \rangle_M \leq 0$ **then return** $-j'$

$p_0 = 0$, $r_0 = j'$, $q_0 = -j'$
loop
  if $\langle Hq_k, q_k \rangle_M \leq 0$ **then return** p_k
  $\alpha_k = \langle r_k, r_k \rangle_M \, / \, \langle Hq_k, q_k \rangle_M$, $\quad p_{k+1} = p_k + \alpha_k q_k \quad$ and $\quad r_{k+1} = r_k + \alpha_k Hq_k$
  if $\|r_{k+1}\|_M < \eta \, \|j'\|_M$ **then return** p_{k+1}
  $\beta_{k+1} = \|r_{k+1}\|_M^2 \, / \, \|r_k\|_M^2 \quad$ and $\quad q_{k+1} = -r_{k+1} + \beta_{k+1} q_k$
end loop

Three possible choices for this sequence have been described and studied by [43]. These three choices were then compared in the context of full waveform inversion in [118], who advise to use the forcing sequence

$$\eta_n = \frac{\|j'(m_n) - j'(m_{n-1}) - \gamma_{n-1} H(m_{n-1}) p_{n-1}\|_M}{\|j'(m_{n-1})\|_M}. \tag{2.38}$$

If the accuracy of the local quadratic approximation is good then this forcing term is close to zero and the Newton system is solved accurately. If not, then iterations are truncated sooner. This forcing sequence plays a similar role than the prospective ratio for trust-region methods. It is however based on a (prospective) expansion of the gradient while the prospective ratio is based on an expansion of the misfit. Additional safeguards are also added to prevent this forcing term to decrease too fast or to increase above $\eta_0 = 0.9$. Interestingly, directions of negative curvatures are never investigated, except if it is the gradient. As previously an appropriate length γ is then given to this direction p through a line search. The unit step length $\gamma = 1$ is again tried first because it would be the best choice if the misfit were quadratic.

Trust region globalization

When the Newton method is associated with a trust-region technique, the direction is found by minimizing the local quadratic expansion of the misfit

$$J^{\text{pred}}(p) := J(m) + \langle j', p \rangle_M + 0.5 \langle Hp, p \rangle_M \tag{2.39}$$

inside a sphere of radius Δ. The constraint $\|p\|_M \leq \Delta$ limits the size of the direction and aims at preventing over-solving. This trust-region sub-problem can be solved approximately with the Steihaug conjugate gradient method (*cf.* Algorithm 6) [173]. This method actually exploits two properties of the conjugate gradient algorithm: successive approximate solutions always grow in norm while the misfit prediction always decreases

$$\|p_k\|_M < \|p_{k+1}\|_M \quad \text{and} \quad J^{\text{pred}}(p_k) > J^{\text{pred}}(p_{k+1}). \tag{2.40}$$

The underlying idea of the method is then to minimize the second order expansion of the misfit iteratively using the conventional conjugate gradient algorithm until either convergence is achieved, or the boundary is reached. Basically, there are only two modifications compared to Algorithm 5. Firstly, the inner iterations are cropped to the trust region radius Δ when the unconstrained solution increases beyond it. Secondly, when a direction of negative curvature is encountered, it is followed up to the boundary of the trust region and the algorithm is stopped. Interestingly these directions are never investigated in the conventional version. The convergence criterion is unchanged but here the forcing term is kept constant ($\eta = 0.5$).

Algorithm 6 Steihaug conjugate gradient algorithm

Require: $\Delta > 0$ and $0 < \eta \leq 1$

$p_0 = 0$, $r_0 = j'$, $q_0 = -j'$
loop
  if $\langle Hq_k, q_k \rangle_M \leq 0$ **then**
    $\tau^* = \tau > 0$ such that $\|p_k + \tau q_k\|_M = \Delta$
    return $p_k + \tau^* q_k$
  end if
  $\alpha_k = \langle r_k, r_k \rangle_M / \langle Hq_k, q_k \rangle_M$
  if $\|p_k + \alpha_k q_k\|_M \geq \Delta$ **then**
    $\tau^* = \tau > 0$ such that $\|p_k + \tau q_k\|_M = \Delta$
    return $p_k + \tau^* q_k$
  end if
  $p_{k+1} = p_k + \alpha_k q_k$ and $r_{k+1} = r_k + \alpha_k H q_k$
  if $\|r_{k+1}\|_M < \eta \|j'\|_M$ **then return** p_{k+1}
  $\beta_{k+1} = \|r_{k+1}\|_M^2 / \|r_k\|_M^2$ and $q_{k+1} = -r_{k+1} + \beta_{k+1} q_k$
end loop

2.3 Reference case studies

Numerical case studies are performed using the least squares distance between the measured and simulated data as a misfit and with the Helmholtz operator as the wave propagation model

$$F_\omega(s^2) := \operatorname{div}(\mathbf{grad}\,()) + \omega^2 s^2. \tag{2.41}$$

For the first two case studies, related to acoustics, the slowness squared distribution s^2 [s^2/km^2], which is the squared inverse of the velocity, is chosen as the subsurface model parameter. Several other parametrizations are also possible but it has been shown that the slowness squared can yield a fast convergence and accurate results [9, 25, 70, 138]. For the third case study, related to electromagnetics, the dimensionless slowness squared s^2 is only an intermediate variable, as the ultimate model parameters are the relative permittivity and the relative conductivity. While relative permittivity is a widely-used concept, that of relative conductivity is less common and it is thus introduced in the subsection related to this third case study.

Implementation of any of the above described local optimization algorithms requires the adjoint state method, developed in Chapter 1, to compute derivatives efficiently. It is summarized here below in terms of a wave field $u_{\omega e}$, a linear forward operator $F_\omega(m)$ and a (perturbed)(adjoint) excitation source $(\delta)f_e^{(\dagger)}$, that can all be identified based on Chapter 1 for the Helmholtz (I) equation and based on Appendix A for the other three wave propagation models.

1. Find the forward fields $u_{\omega e}$ such that

$$F_\omega(m)u_{\omega e} = f_e. \tag{2.42}$$

2. Find the adjoint fields $u^\dagger_{\omega e}$ such that

$$F_\omega(m)u^\dagger_{\omega e} = f^\dagger_e. \tag{2.43}$$

3. Find the preconditioned gradient j' such that

$$Pj' = j'_{L_2}. \tag{2.44}$$

4. Find the perturbed forward fields $\delta u_{\omega e}$ such that

$$F_\omega(m)\delta u_{\omega e} = \delta f_e. \tag{2.45}$$

5. Find the perturbed adjoint fields $\delta u^\dagger_{\omega e}$ such that

$$F_{\omega,e}(m)\delta u^\dagger_{\omega e} = \delta f^\dagger_e. \tag{2.46}$$

6. Find the preconditioned Hessian operator $H\delta m$ in the direction δm such that

$$PH\delta m = \delta j'_{L_2}. \tag{2.47}$$

Independently of any practical solver for these wave propagation problems, a misfit evaluation only requires performing step 1 and thus only requires solving a single wave propagation problem. A gradient evaluation requires steps 1 to 3, thus a single supplementary wave propagation problem must be solved if the misfit has already been computed. Similarly, steps 1 to 6 are necessary for the application of the (Gauss-)Newton Hessian operator in a particular direction, thus again two supplementary wave propagation problems if the gradient has already been computed for the same model parameters.
Consequently the steepest descent and the l-BFGS directions require solving two wave propagation problems while any Newton-based direction has an initial cost of four wave problems and each supplementary conjugate gradient iteration requires two more wave problems. To the price of the directions must be added the cost of the line search or the trust-region methods. Line search typically accepts a step length if it verifies sufficient conditions (2.4) and (2.5), which involves the misfit and its gradient. Thus it requires one or two additional wave problems each time a trial step length is rejected. Prospective trust-region has no additional cost because the evaluation of the trust region only depends on quantities already computed.

	Base	CG	LS	TR
SD	2	-	$2N_{\text{LS}}$	0
l-BFGS	2	-	$2N_{\text{LS}}$	0
LS-NCG	2	$2N_{\text{CG}}$	$2N_{\text{LS}}$	-
TR-NCG (P)	2	$2N_{\text{CG}}$	-	0
TR-NCG (R)	2	$2N_{\text{CG}}$	-	2

Table 2.2: Wave propagation problem solutions count for a single outer iteration of each local optimization algorithm. N_{CG} is the number of inner iteration of the conjugte gradient algorithm. N_{LS} is the number of rejected values of γ during the line search.

At the opposite, retrospective (Gauss-)Newton trust-region requires the application of the Hessian operator on the preceding direction and thus needs to solve two additional wave propagation problems. Table 2.2 summarizes this accounting. It is interesting to highlight here that the first inner iteration of any conjugate gradient Newton method is simply the steepest descent but it is twice more expensive because the curvature is computed. Subsequent inner iterations must therefore provide large decrease of the misfit to compensate this high entry cost. This phenomenon is even worse with the retrospective trust-region algorithm because there is a systematical additional cost to update the trust region radius.
Solutions to partial differential equations (2.42) to (2.47) are obtained numerically with the finite element method. In what follows, the exact numerical procedure is specified in that context. Note however that the analysis would nearly be identical with finite differences. Finite element discretization assembles operators into matrices and source terms into vectors. Wave propagation problems (2.42), (2.43), (2.45) and (2.46) therefore transform into a linear system whose left-hand-side matrix F is always the same for a given frequency while the right-hand-side source f is different for any field type, frequency and excitation index. The solution of this system is obtained by first computing its lower-upper factorization then by performing an upward-backward substitution for each right-hand-side source

$$\mathsf{Fu} = \mathsf{f} \quad \Leftrightarrow \quad \mathsf{F} = \mathsf{LU}, \quad \mathsf{Lv} = \mathsf{f} \text{ and } \mathsf{Uu} = \mathsf{v}. \tag{2.48}$$

Huge computational reduction is therefore obtained because only one matrix per frequency is assembled and factorized. The computation of any wave field then requires the assembly and the upward-backward substitution of a vector per excitation source, but no more matrix factorization. The numerical equivalence of the preceding six steps procedure is given here below.

1. • Factorize wave propagation operators (n_ω)

	• Substitute forward sources	$(n_\omega \times n_e)$
2.	• Substitute adjoint sources	$(n_\omega \times n_e)$
3.	• Factorize the preconditioner	(1)
	• Substitute the conventional gradient	(1)
4.	• Substitute perturbed forward sources	$(n_\omega \times n_e)$
5.	• Substitute perturbed adjoint sources	$(n_\omega \times n_e)$
6.	• Substitute the conventional Hessian	(1)

It is interesting to highlight that model problems (steps 3 and 6) are negligible with respect to wave problems. Indeed while wave problems involve a matrix per frequency and a vector per excitation source, model problems only involve a single matrix (the preconditioner) and a single source vector (the conventional gradient or Hessian). Moreover, the model discretization is usually coarser than the wave field discretization. Consequently, not considering these model problems when quantifying the computational complexity is not dramatic. It should however be highlighted that forward problems are more expensive than the corresponding adjoint problem, because the matrix factorization is reused. Moreover, the perturbed forward and the perturbed adjoint problems are also slightly more expensive than the adjoint problem, because both their sources δf_e and $\delta f_e^\dagger$ are dense, meaning that both are non-vanishing everywhere in the domain Ω. Consequently, each entry of the corresponding discrete source vector must be computed. At the opposite, the forward and adjoint sources are sparse, as both are only defined on emitters and receivers respectively, and only a small number of entries then needs to be computed. Nevertheless, all of these four problems are weighted equally when quantifying the computational complexity.

In the next three sections, the three reference case studies are investigated. As a reminder, the first one is based on the widely used Marmousi benchmark [187] while the second one, replicated from [118], is inspired from a near-surface imaging of close concrete structures and features important multiple scattering. Multiple scattering is responsible for the indefiniteness of the Hessian operator, which, as mentioned in Section 2.2, is challenging for optimization algorithms. This second example is thus chosen to emphasize which optimization methods are able to overcome such difficulties. Finally, the third test case, replicated from [97, 111], is considered. The particularity of this last case study is to feature two parameters. Multi-parameter inversions are known to be challenging because of the trade-offs between parameters, that can typically be captured with the Hessian operator.

For all these case studies, the influence of the inner product choice on the convergence speed and the quality of the inverted model is studied first. Once the inner product is chosen, prospective and retrospective trust-region methods with different parameter sets are compared

and the best option is selected. Advantages and drawbacks of trust-region methods in the context of full waveform inversion are then finally discussed based on a comparison with the corresponding line search methods. In the remainder of this chapter, data misfit are normalized such that the misfit corresponding to the initial model is one and computational complexity is measured in numbers of forward problems solved, as explained above.

2.3.1 Case study 1: Marmousi model

Numerical inversions are performed on the 2D Marmousi model (*cf.* Figure 2.4a) with three frequencies (4, 6 and 8 [Hz]) simultaneously, starting from a smoothed version of the exact Marmousi model (*cf.* Figure 2.4b). Outer iterations are stopped when satisfying the convergence criterion $J(s^2)/J(s^2_{\text{init}}) < 10^{-3}$. An inversion result, *i.e.* an estimated squared slowness, is shown in Figure 2.4c. From a relatively low resolution initial guess, full waveform inversion indeed provides a high-resolution estimation of the exact model. Images obtained with the other methods do not differ significantly.

Inner product

As explained earlier, the inner product has an influence on both the gradient and the Hessian. Its choice is therefore expected to influence the convergence speed but also the particular minimizer that is reached. To illustrate both these effects, the line search *l*-BFGS algorithm has been applied with the four different inner products introduced in this chapter, *i.e.* the conventional inner product (2.19), the weighted inner product (2.20) and its regularized variants (2.22) or (2.23). Corresponding convergence curves and error maps are given in Figure 2.5 and 2.6 respectively. Both these figures are also summarized in Table 2.3. As can be seen from these figures, all weighted inner product increase the convergence speed with respect to the conventional, *i.e.* unweighted, one. However, the minimizer obtained with the weighted inner product alone is further away from the exact solution, in particular in the right corner of the model. Avoiding such artifacts is precisely one of the reasons for the introduction of regularized inner products, as they dampen the contributions in these poorly illuminated regions. Both the thresholding and the smoothing strategy perform similarly in reducing the error back to the same level than the unweighted solution but the thresholding strategy converges faster. It is thus kept for the sequel of this case study. The advantages of the smoothing inner product will be highlighted during the second case study.
In the next three subsections, the behaviour of the steepest descent method, the *l*-BFGS method, the full Newton and the Gauss-Newton methods is analysed. Convergence curves and interesting statistics for all these methods are given in Figure 2.7 and Table 2.4 respectively.

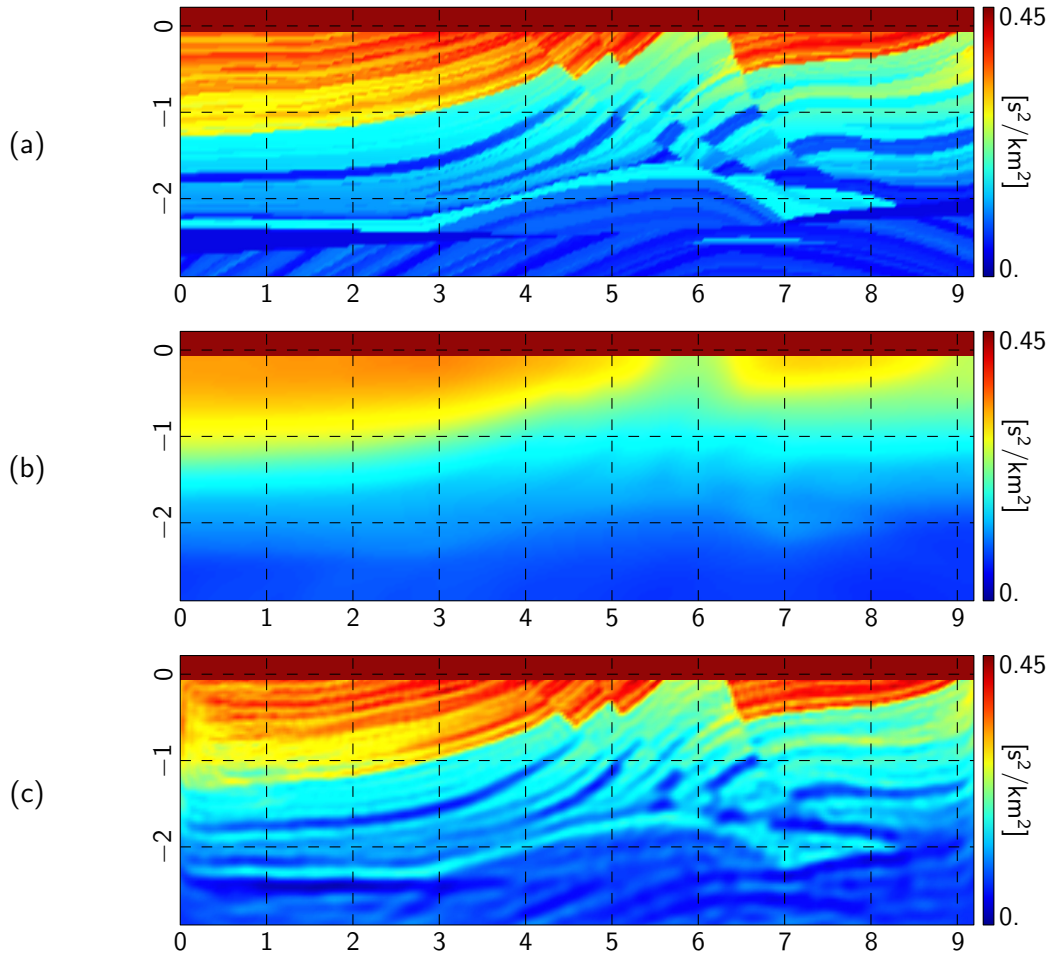


Figure 2.4: Marmousi slowness squared model (a), initial guess (b) and inversion results using a line search *l*-BFGS algorithm with a weighted and thresholded inner product (c).

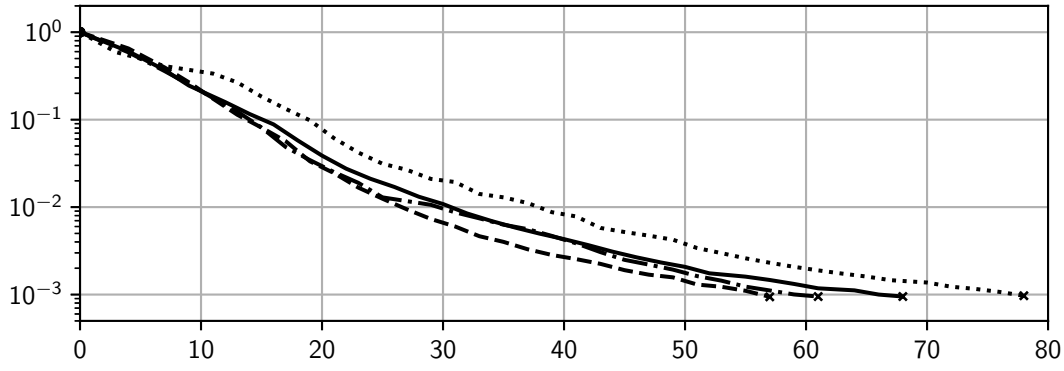


Figure 2.5: Data misfit as a function of the computational complexity for the line search *l*-BFGS algorithm with a conventional (••), only weighted (-•), weighted and thresholded (- -) or weighted and smoothed (–) inner product.

		Wave sol. (tot)	Error rms ([s^2/km^2])
Conventional		78	0.0174
Weighted	only	61	0.0202
	and thresholded	57	0.0174
	and smoothed	68	0.0173

Table 2.3: Computational complexity and root-mean squared error for the line search *l*-BFGS algorithm with different inner products.

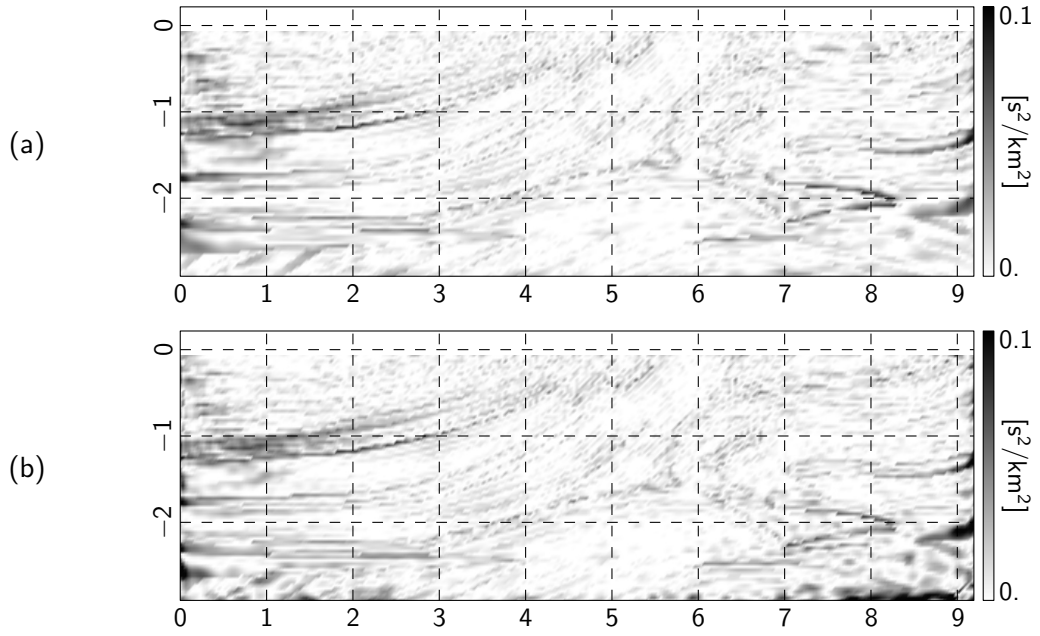


Figure 2.6: Final inversion error for the line search *l*-BFGS algorithm with a conventional (a) or a weighted (b) inner product. Inversion errors for both regularized inner products are not shown because these are very similar to those obtained with the conventional inner product.

		Wave sol. (tot)	Outer it. (tot)	Inner it. (avg)	Rejected (%)	Constrained (%)	Negative curv. (%)
SD	LS	244	111	10	-	-	-
	TR-P (A)	396	198	-	40	100	-
	TR-P (B)	280	140	-	6	100	-
	TR-P (C)	264	132	-	5	100	-
	TR-R (A)	328	164	-	20	100	-
	TR-R (B)	354	177	-	20	100	-
	TR-R (C)	330	165	-	25	100	-
LB	LS	57	27	-	7	-	-
	TR-P (A)	58	29	-	3	10	-
	TR-P (B)	58	29	-	3	34	-
	TR-P (C)	64	32	-	13	50	-
	TR-R (A)	58	29	-	3	10	-
	TR-R (B)	56	28	-	0	11	-
	TR-R (C)	56	28	-	0	11	-
FN	LS	139	17	2.9	12	-	29
	TR-P (A)	178	22	3.0	32	77	0
	TR-P (B)	106	13	3.1	0	69	0
	TR-P (C)	106	16	2.3	0	75	0
	TR-R (A)	144	14	3.1	14	64	0
	TR-R (B)	128	14	2.6	0	79	0
	TR-R (C)	142	17	2.2	0	82	0
GN	LS	124	15	3.13	0	-	-
	TR-P (A)	130	11	4.9	0	10	-
	TR-P (B)	98	10	3.9	0	30	-
	TR-P (C)	98	10	3.9	0	30	-
	TR-R (A)	152	11	4.9	0	10	-
	TR-R (B)	132	14	2.7	0	79	-
	TR-R (C)	184	24	1.8	0	83	-

Table 2.4: Statistics related to the implementation of the steepest descent (SD), the *l*-BFGS (LB), the full Newton (FN) method and the Gauss-Newton (GN) methods combined with a line search (LS) or a trust region (TR) with a prospective (P) or retrospective (R) radius update with different parameter sets (A,B,C).

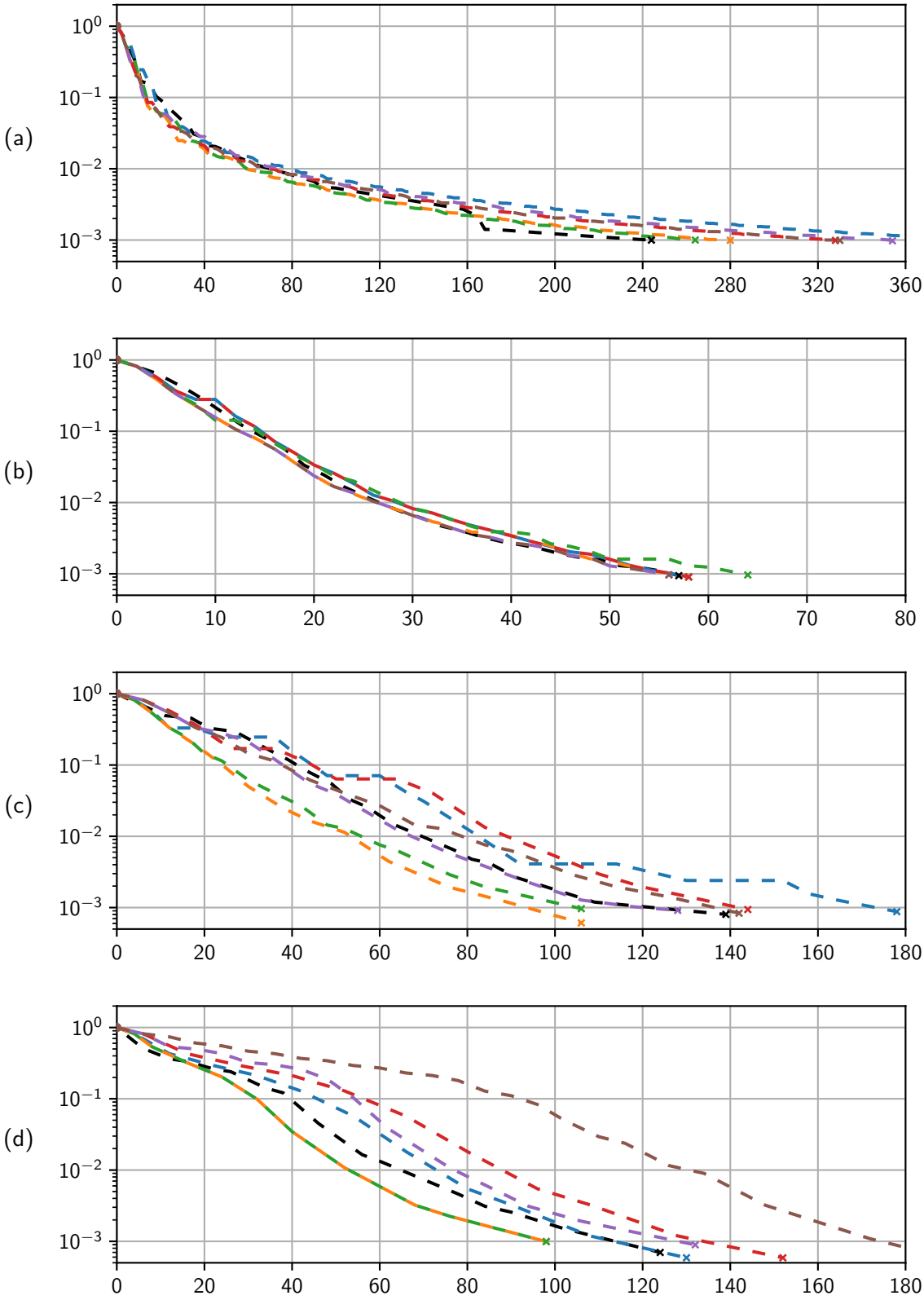


Figure 2.7: Data misfit as a function of the computational complexity for the steepest descent (a), the *l*-BFGS (b), the full Newton (c) and the Gauss-Newton (d) methods combined with either a line search (—) or a prospective trust region (A (—), B (—), C (—)) or a retrospective trust region (A (—), B (—), C (—)).

Steepest descent

There is no dramatic improvement when using one or another direction scaling method, because actually the direction itself is bad. Nevertheless, it appears that methods which reject less frequently the proposed update direction are faster, *i.e.* the prospective trust-region method with the more cautious parameters sets (B and C) and the line search method. Retrospective radius update does not speed up convergence. Actually the retrospective predicted decrease (2.15) sometimes largely underestimates the actual decrease, illustrating that the retrospective misfit prediction is very not accurate, but still producing an increase of the trust region radius. Finally, among the three best methods, the slope is slightly steeper for the two trust-region methods, probably because they systematically try to increase the length given to the gradient direction.

Limited memory BFGS method

There is hardly any difference between all the methods combined with the *l*-BFGS algorithm. The line search method only rejects the unit step length $\gamma = 1$ for the first two iterations. Similarly, the retrospective ratio is always very close to one, such that the trust region radius for retrospective methods quickly becomes large and thus the pure (unconstrained) *l*-BFGS direction is always accepted after the first few iterations.
An algorithm that unconditionally follows the pure *l*-BFGS direction would therefore already be very good and neither a line search nor a trust-region method can actually drastically improve it, as far as convergence speed is concerned. Nevertheless, the more cautious prospective trust-region methods (B and C) also converge fast, which shows that, on the other hand, constraining the size of the update directions does not slow down the inversion.

Newton methods

As far as trust-region methods are concerned, it first clearly appears that the retrospective radius update is not worth its computational cost. Indeed, it does not require less wave solutions than the best prospective ones, even if the computation cost of the retrospective predicted decrease is withdrawn (two wave solutions per outer iteration). Retrospective radius update has been introduced to anticipate and prevent failures. However, the prospective Newton method combined with the more cautious parameters sets (B and C) does already not reject any direction and there is then no interest in computing the retrospective ratio.
Among the prospective methods, it appears that the two more cautious (B and C) yield the fastest convergence. Convergence speed decreases when using parameter set A with both the full Newton method and the Gauss-Newton method but for two different reasons. With parameter set A, the trust region radius grows quickly and the full Newton method is thus allowed to explore large directions, beyond the validity of the exact second order

expansion (2.2). Such directions produce a high rejection rate (32%) and thus a waste of computational effort. At the opposite, the Gauss-Newton method never rejects a direction and the explanation for its slower convergence can therefore not be the same. During the earliest iterations, far from the global minimum, the Gauss-Newton approximation is not valid (because data residuals are not small yet) and thus the Gauss-Newton Hessian is quiet different from the full Hessian. The misfit prediction under the Gauss-Newton approximation is thus cruder than the exact second order expansion (2.2) and the ratio ρ_p is even more likely to be away from one. This ratio ρ_p is given in Figure 2.8c. As can be seen, during the first few iterations, this ratio is actually very larger than one, which indicates that the misfit prediction is indeed not accurate. Nevertheless, the trust region radius is still increased and the system is solved accurately while the Hessian and the misfit are not approximated accurately. This effect generates over-solving the system at the earliest iteration and slows down the Gauss-Newton method, as can be seen by comparing the initial slopes between variant A and B/C in Figure 2.7d. This effect would be even more dominant if the convergence requirements, *i.e.* the forcing sequence η, were smaller. With the large value $\eta = 0.5$ chosen here, convergence of the conjugate gradient algorithm is attained relatively fast. Actually variant B and C perform better than variant A only because it takes more iterations for the trust region constraint to become inactive. Starting with a larger initial radius would result in the same convergence speed than variant A. Also, it is interesting to highlight that when using the retrospective radius update with the Gauss-Newton approximation, the situation is reversed because the retrospective ratio is then smaller than one. Instead of over-solving, under-solving then appears. It is therefore better to use trust-region methods with the full Newton Hessian, because it constructs the best possible misfit prediction while it does not introduce supplementary difficulties.

The full Newton method and the Gauss-Newton method are slightly slower when combined with a line search method. As far as the full Newton method is concerned, directions of very small curvature can produce large update directions, far beyond the validity of the expansion (2.2). In such cases, the initial length $\gamma = 1$ is rejected and some computational cost must be involved to reduce it to satisfy Wolfe conditions. This effect has actually been observed twice using the full Newton method. Moreover during the first fifth outer iterations, the full Newton method using the line search globalization stops because a direction of negative curvature is encountered. At the opposite of its trust-region counterpart, the line search variant of the conjugate gradient algorithm discard any direction of negative curvature, thus wasting the associated computational cost. Of course within the Gauss-Newton approximation this second effect can not appear (and the first one was actually not observed). The line search globalization therefore seems more suited with the Gauss-Newton approximation. Nevertheless it is not much faster. In the context of line search globalization, the accuracy of the second order local expansion is expressed through the forcing sequence η, which is, as can be seen in Figure 2.10, away from zero during the first few iterations. Consequently,

the convergence of the conjugate gradient algorithm is quickly reached and only a few inner iterations are performed per outer iteration, as can be seen from Figure 2.9c. Figure 2.9c actually show how hard it is to stop the Gauss-Newton inner iterations at the right time: the fastest method is the prospective trust region B or C and it performs less inner iterations then the variant A but more than the line search method. The difficulty to pick up an appropriate stopping criterion for the Gauss-Newton method is another motivation to use the full Newton method instead. Using the full Newton method, the line search variant actually suffers from directions of small or negative curvature while trust-region methods do not. Based on this first case study, the recommendation is therefore to use the full Newton method combined with a trust-region method and a prospective radius update.

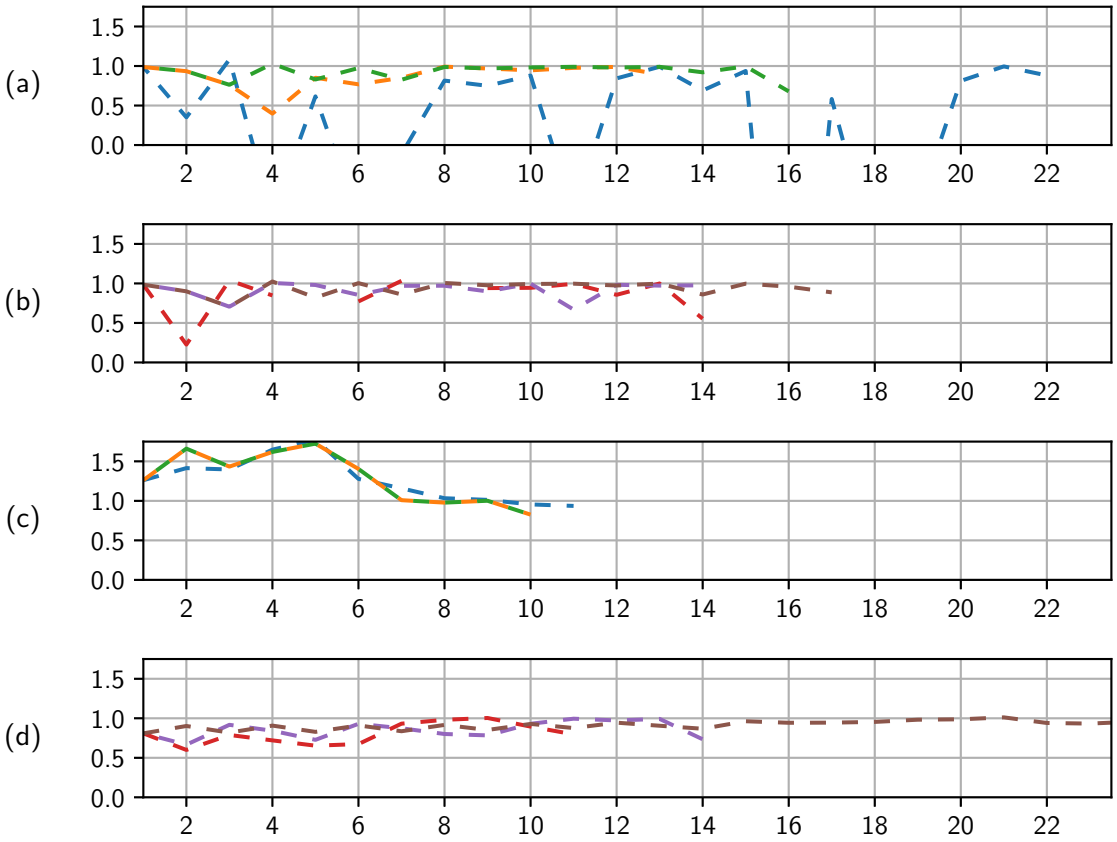


Figure 2.8: Prospective ratio ρ_p (a,c) or retrospective ratio ρ_r (b,d) during the outer iterations of the full Newton method (a,b) and the Gauss-Newton method (c,d) with different parameter sets using a prospective radius update (a,c) (A (—), B (—), C (—)) or a retrospective radius update (b,d) (A (—), B (—), C (—)).

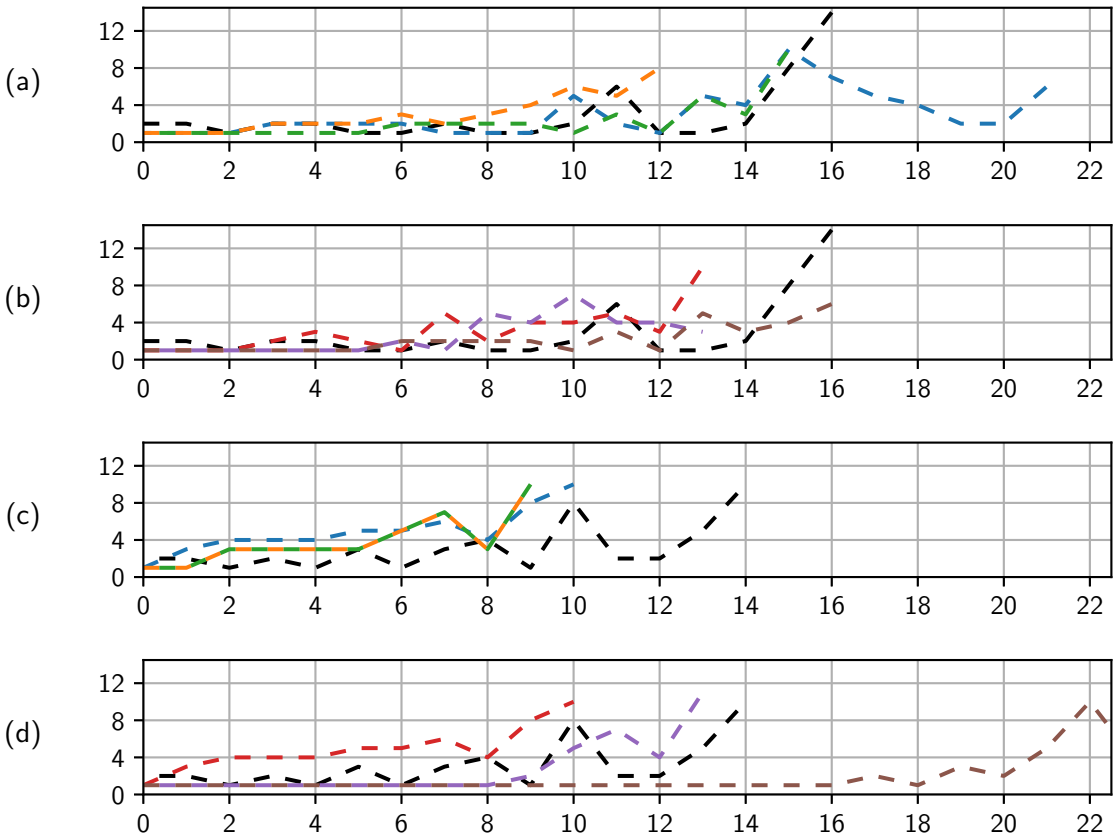


Figure 2.9: Inner iterations per outer iteration for the full Newton method (a,b) and the Gauss-Newton method (c,d) with different parameter sets using a prospective radius update (a,c) (A (▬), B (▬), C (▬)) or a retrospective radius update (b,d) (A (▬), B (▬), C (▬)).

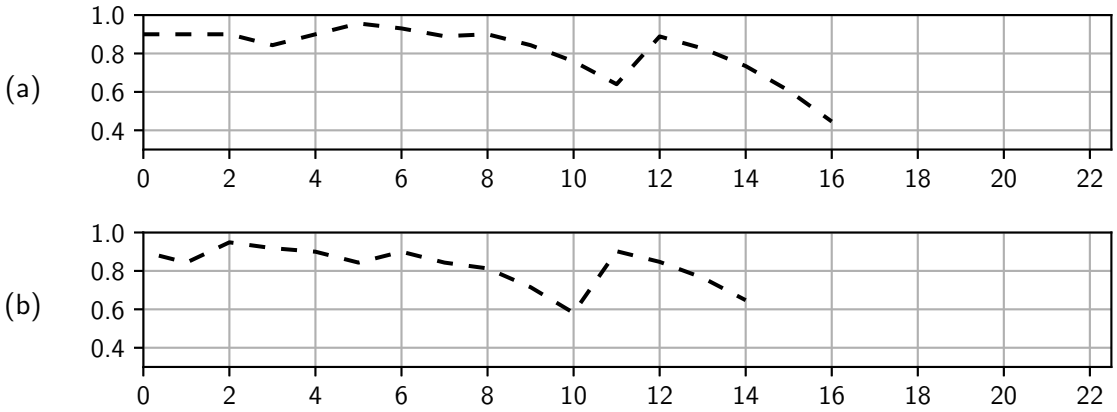


Figure 2.10: Forcing sequence η for the full Newton (a) and the Gauss-Newton (b) methods combined with a line search method (▬). The forcing sequence for methods combined with a trust-region is constant ($\eta = 0.5$).

2.3.2 Case study 2: T-shaped reflectors

The true velocity distribution of this second case study is given in Figure 2.11a. As a reminder, two highly-contrasted T-shaped concrete structures ($v_c = 4$ [km/s]) are embedded in a homogeneous background ($v_b = 0.3$ [km/s]) with a bottom reflector ($v_r = 0.5$ [km/s]). These two concrete foundations generate high-amplitude reflections and in particular, important multiple scattering between the two structures. The acquisition system is divided into three segments and nine frequencies (100, 125, 150, 175, 200, 225, 250, 275, and 300 [Hz]) are inverted simultaneously from an empty initial model (Figure 2.11b). For this second case study, a logarithmic slowness squared parametrization is used $m := \ln s^2$. This parametrization has the advantage to be unable to produce negative values of the slowness squared. Inverting the slowness squared actually drives it into negative values, because of the two concrete structures whose slowness squared is really close to zero. This logarithmic slowness squared field is discretized by hierarchical finite elements of order 1. Outer iterations are stopped when satisfying the convergence criterion $J(\ln s^2)/J(\ln s^2_{\text{init}}) < 10^{-2}$. At the light of the first case study, trust-region methods with parameter sets A and C will no longer be considered, as both were systematically outperformed by parameter set B.

Inner product

Illumination of the medium is nearly perfect and consequently, the diagonal part of the Gauss-Newton Hessian that has previously been used as a weight can reasonably be approximated by a constant h_{GN}. However, the part related to the change of variable is varying spatially

$$\delta s^2 = \frac{ds^2}{d\ln s^2}\,\delta \ln s^2 = s^2\,\delta \ln s^2. \tag{2.49}$$

Hence the weight for the inner product is chosen as

$$w = h_{\text{GN}}\; s^4. \tag{2.50}$$

As previously, the line search *l*-BFGS algorithm has been applied with the four different inner products introduced in this work. Inversion velocity results are given in Figure 2.11 while convergence curves are given in Figure 2.12. For the weighted and smoothed variant, the threshold is set to $\epsilon = h_{\text{GN}}\; s_b^4$ while the characteristic length for the smoothing inner product is set to $l_c = 3$ [m]. It is important to highlight that this length is greater than the smallest wavelength in the background medium (1 [m]) while for the first case study, this length was actually close to the smallest wavelength. The weighted and thresholded variant has been tested for several values of the threshold, from $\epsilon = h_{\text{GN}}\; s_c^4$ to $\epsilon = h_{\text{GN}}\; s_b^4$ but none of them provided inversion results significantly different from the conventional or the weighted-only inner products. Only the smoothing inner product is able to reconstruct the model parameter

accurately. This smoothing inner product actually mitigates the non-linearity of the misfit, because spatial roughness is incorporated progressively in the model parameter [210]. During the inversion, the model parameter never explores extremely high velocity values, at the opposite of the other variants. It is thus able to converge to an accurate solution while more straightforward optimization is not. Consequently, this inner product is used for the remainder of this study.

The performance of the three optimization methods is described in the next three subsections. Inversion velocity results, convergence curves and statistics are given in Figure 2.13 and 2.14 and in Table 2.5 respectively.

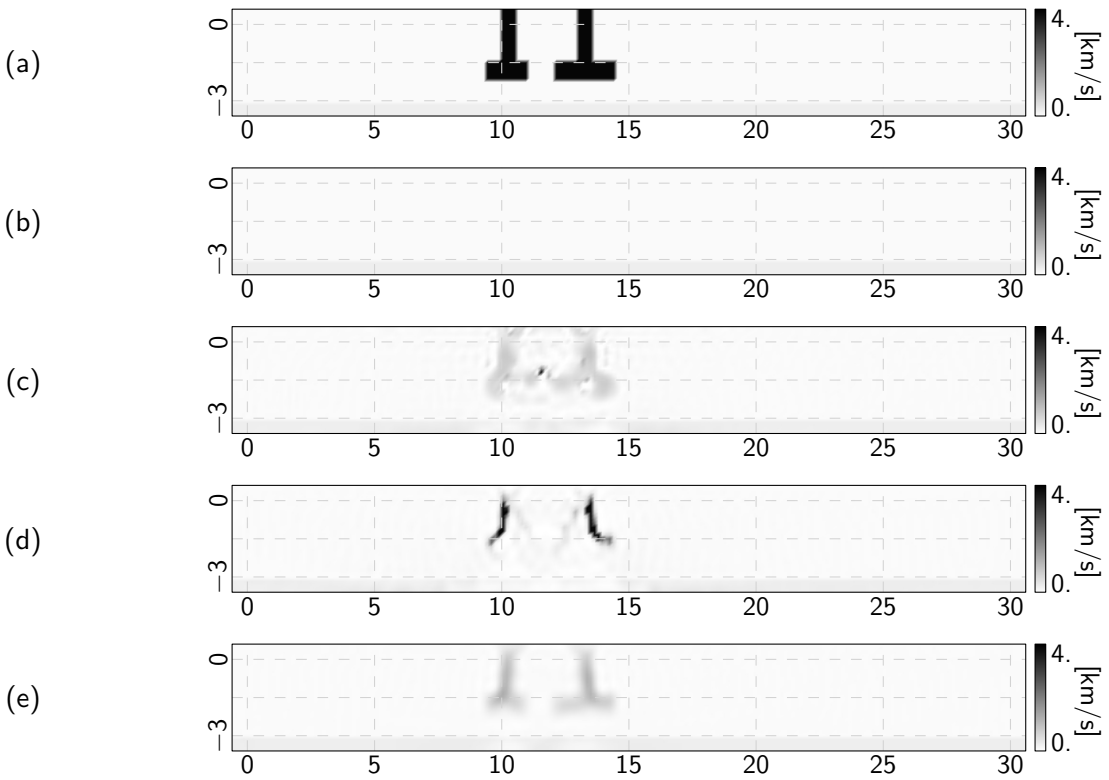


Figure 2.11: Near-surface concrete structures velocity model (a), initial guess (b) and inversion results using a line search *l*-BFGS algorithm with a conventional (c), a weighted-only (d) or a weighted and smoothed (e) inner product.

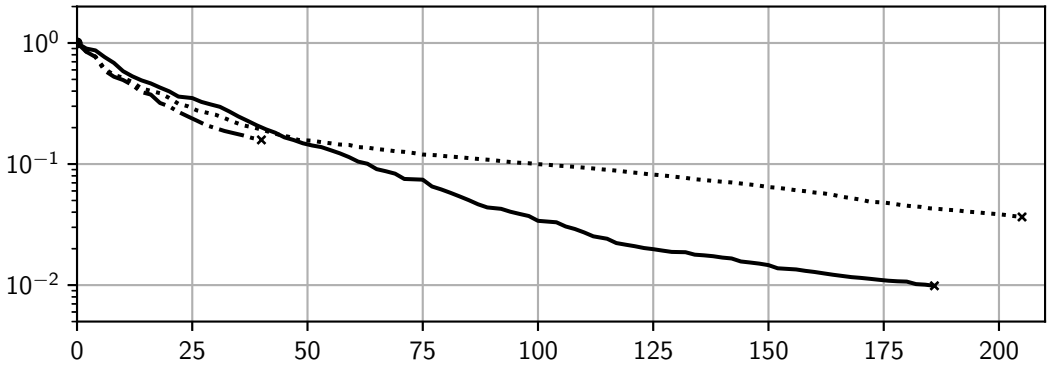


Figure 2.12: Data misfit as a function of the computational complexity for the line search *l*-BFGS algorithm with a conventional (••), weighted-only (-•) or weighted and smoothed (–) inner product.

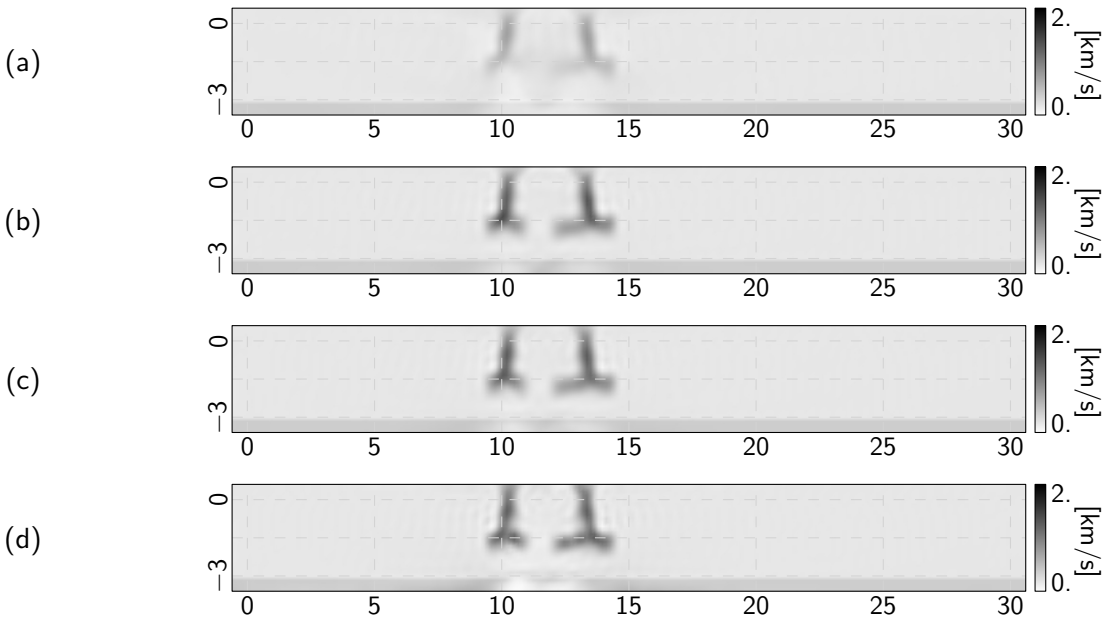


Figure 2.13: Inversion velocity results for the steepest descent (a), the *l*-BFGS (b), the full Newton (c) and the Gauss-Newton (d) methods combined with trust-region method using a prospective radius update (B). Note that the upper color scale limit is only 2 [km/s].

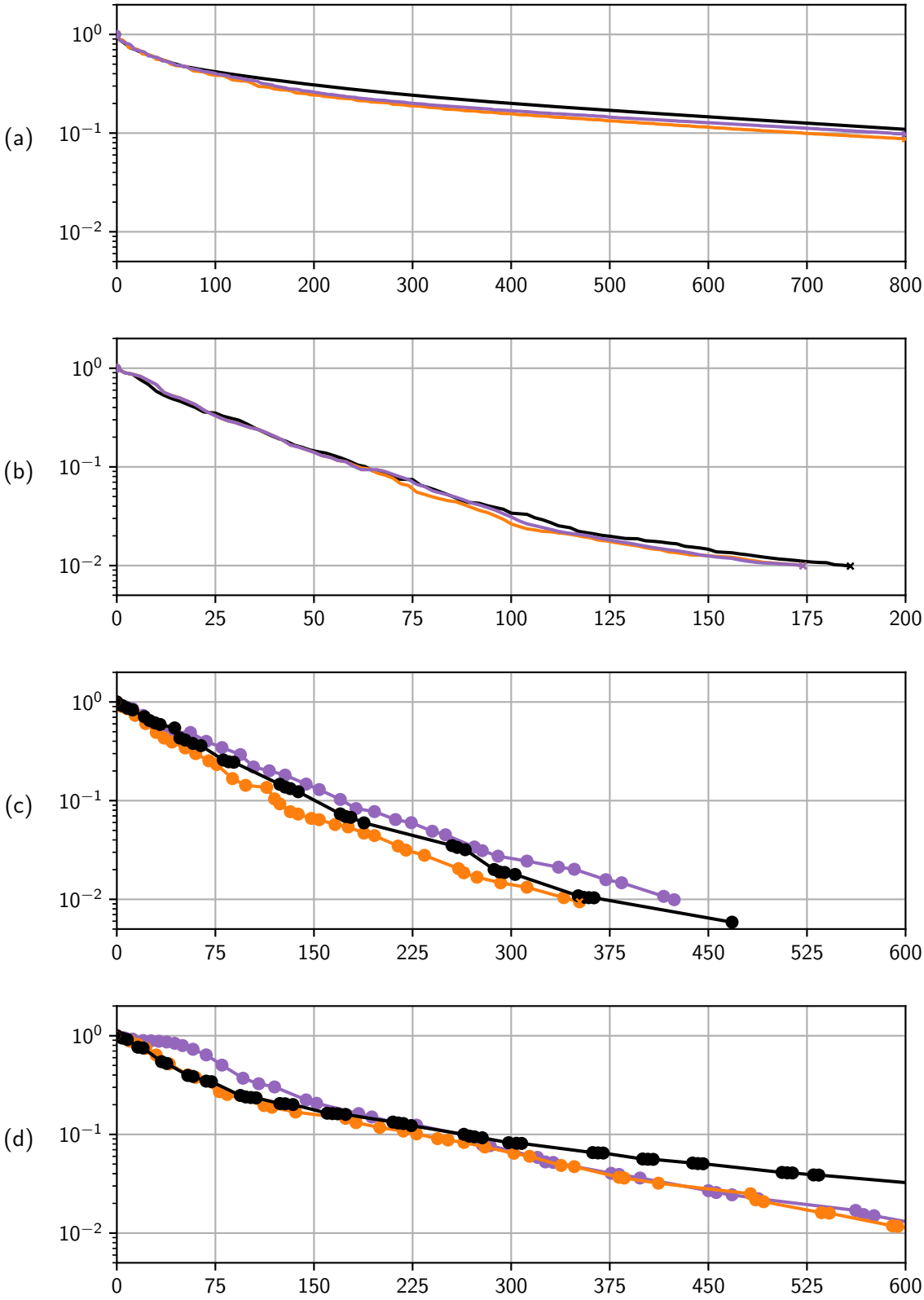


Figure 2.14: Data misfit as a function of the computational complexity for the steepest descent (a), the *l*-BFGS (b), the full Newton (c) and the Gauss-Newton (d) methods combined with either a line search (▬) or a trust-region with a prospective (B (▬)) or a retrospective (B (▬)) radius update. Dots on (Gauss)-Newton curves indicate outer iterations.

		Wave sol. (tot)	Outer it. (tot)	Inner it. (avg)	Rejected (%)	Constrained (%)	Negative curv. (%)
SD	LS*	803	400	-	1	-	-
	TR-P (B)*	800	400	-	9	100	-
	TR-R (B)*	800	400	-	18	100	-
LB	LS	186	88	-	8	-	-
	TR-P (B)	174	87	-	1	23	-
	TR-R (B)	174	87	-	2	6	-
FN	LS	468	35	5.3	23	-	17
	TR-P (B)	352	34	4.2	0	56	0
	TR-R (B)	424	31	4.8	3	68	0
GN	LS*	923	60	6.6	8	-	-
	TR-P (B)	680	38	7.9	0	42	-
	TR-R (B)	672	39	6.6	0	41	-

Table 2.5: Statistics related to the implementation of the steepest descent (SD), the *l*-BFGS (LB), the full Newton (FN) and the Gauss-newton (GN) methods combined with a line search (LS) or a trust-region (TR) with a prospective (P) or retrospective (R) radius update with parameter set B. Star marker (∗) indicates methods that have been stopped before convergence.

Steepest descent

The steepest descent method is not able to reach convergence in a reasonable amount of computations. Progressively decreasing the smoothing length l_c during the inversion would accelerate the convergence [210], but it is not needed for more sophisticated methods and thus it is not done here neither. As for the first test case, the slope of trust-region methods is slightly steeper than the line search method. The prospective radius update rejects less often directions and hence converges faster than the retrospective radius update.

Limited memory BFGS method

Similarly to the first test case, the influence of the globalization method on the convergence speed is small. Trust-region methods actually spare a part of the line search cost, but it already represents only a tiny fraction (20 wave solutions) of the overall computational cost (186 wave solutions). Retrospective ratio is again always very close to one and the only difference

between retrospective and prospective radius update is the frequency the size constraint is active, although it does not influence the convergence speed.

Newton methods

For this case study, the full Newton method clearly outperforms the Gauss-Newton method, independently of the globalization method used. On the one hand, the convergence speed is much higher and on the other hand the accuracy of the inversion results is superior. As demonstrated in [118], the missing negative definite part of the Hessian can prevent the Gauss-Newton method from reaching convergence. Here, thanks to the inner product preconditioning, every method is able to find the minimum but the invalidity of the Gauss-Newton approximation impacts the convergence speed and the inversion results. This is particularly true for the line search method, which has a very slow convergence. Its forcing sequence actually remains close to one even at the end of the inversion (*cf.* Figure 2.17b), because the Gauss-Newton Hessian is not accurate. Hence the system is never solved accurately. Interestingly, for the Gauss-Newton method, the retrospective radius update succeeds to compensate its cost (2 wave solutions per outer iteration). Indeed, during the earliest outer iterations, when the Gauss-Newton and the full Hessian are different, the retrospective ratio is smaller than one while the prospective ratio is bigger than one (*cf.* Figure 2.15b). Consequently, the retrospective method performs less inner iterations per outer iterations than the prospective method (*cf.* Figure 2.16b), and thus avoids early over-solving. In the end, both methods still converge at the same speed, but the retrospective method has spent less time in the computation of linear system solutions (680 versus $672 - 2 \times 39 = 594$ wave solutions). At the opposite, for the full Newton method, the retrospective method spent even more time in the computation of linear system solutions than the prospective method. The prospective method is actually already efficient because the prospective misfit prediction is accurate. The line search globalization also provides fast convergence in this case, despite the fact that directions of negative curvature are often encountered (12 wasted wave solutions) and that the unit step length is often rejected. However, the flow of the method is very different from trust-region methods. Indeed, line search methods have a tendency to compute a single very accurate system solution, followed by several very inaccurate system solutions as can be seen from Figure 2.16a and from the dots spacing in Figure 2.14c, while trust-region methods perform a nearly steadily increasing number of inner iterations per outer iteration. Whether a flow is better than the other has not been emphasized by this case study.

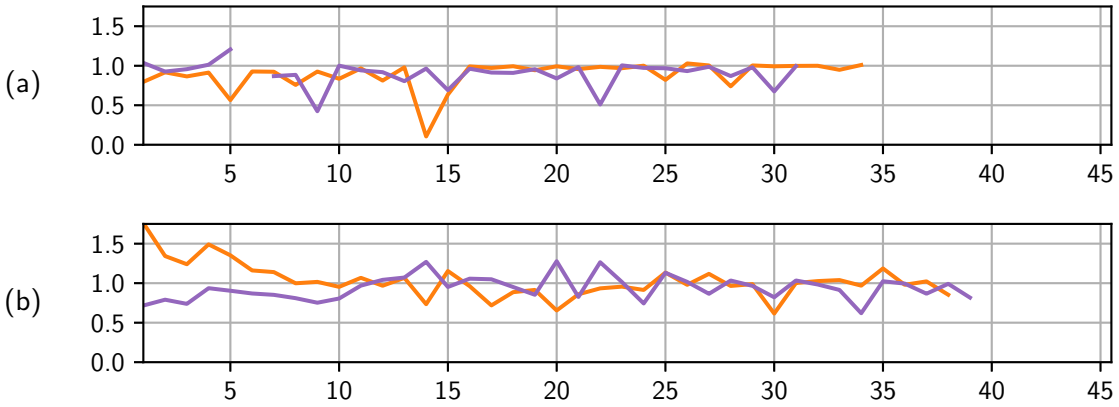


Figure 2.15: Prospective ratio ρ_p using a prospective radius update (B (▬)) or retrospective ratio ρ_r using a retrospective radius update (B (▬)) during the outer iterations of the full Newton method (a) and the Gauss-Newton method (b).

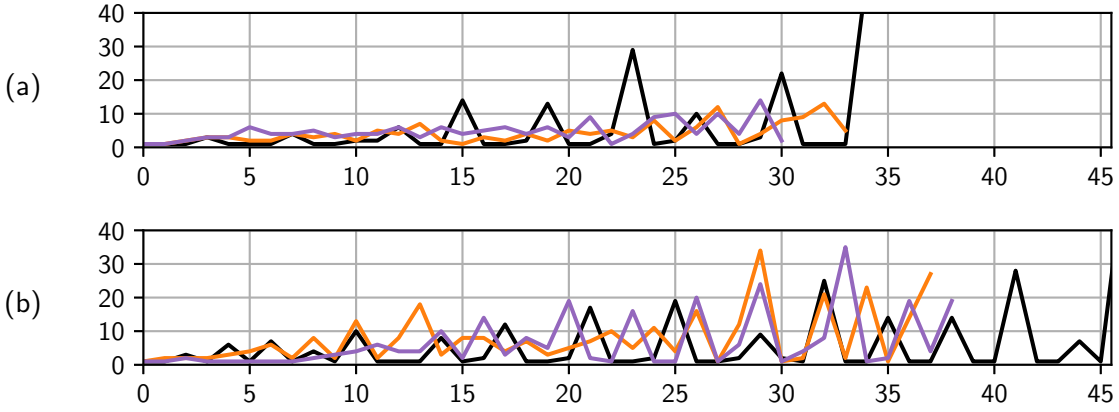


Figure 2.16: Inner iterations per outer iteration for the full Newton method (a) and the Gauss-Newton method (b) combined with either a line search (▬) or a trust-region with a prospective (B (▬)) or a retrospective (B (▬)) radius update.

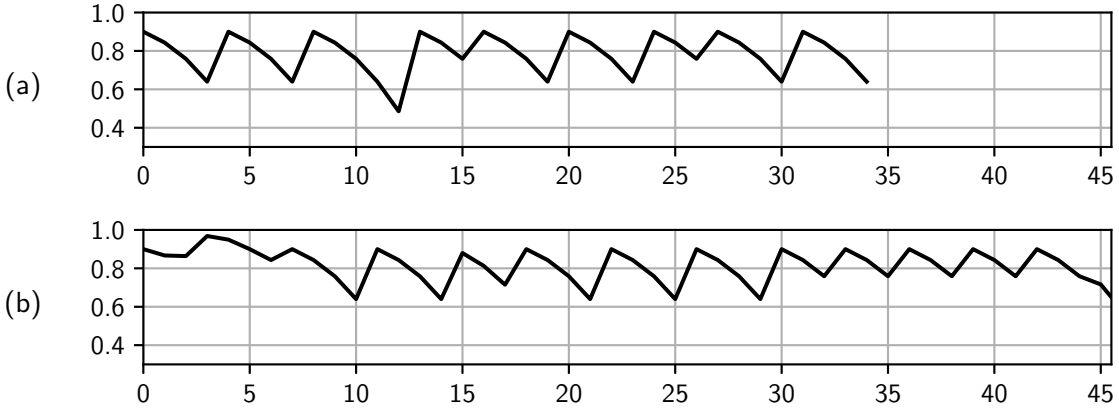


Figure 2.17: Forcing sequence η for the full Newton (a) and the Gauss-Newton (b) methods combined with a line search method (▬). The forcing sequence for methods combined with a trust-region is constant ($\eta = 0.5$).

2.3.3 Case study 3: Dissipative crosses

The ground-truth relative permittivity and conductivity distributions for this third case study are given in Figure 2.19a and 2.19b respectively. As a reminder, they consist in two highly contrasted cross-shaped structures embedded in a homogeneous dissipative background ($\epsilon_{r,0} = 4.$, $\epsilon_{r,1} = 1.$, $\epsilon_{r,2} = 8.$ [-] and $\sigma_0 = 3.$, $\sigma_1 = 0.1$, $\sigma_2 = 10.$ [mS/m]). Seven frequencies (50, 60, 70, 80, 100, 150, 200 [MHz]) are inverted simultaneously from an homogeneous background. Similarly to [97], the relative conductivity σ_r is defined such that the imaginary part of the dimensionless slowness squared at the reference wavenumber k_0 equals one when the relative conductivity also equals one, *i.e.* $\sigma_{\text{ref}} := k_0/\eta_0$ and $\sigma = \sigma_r \sigma_{\text{ref}}$. Therefore, the dimensionless slowness squared can finally be written in terms of the relative permittivity ϵ_r and the relative conductivity σ_r as

$$s^2 := \epsilon_r - i\frac{\sigma_r}{k/k_0}. \tag{2.51}$$

These relative parameters are *nearly* chosen for the model parameters: a logarithmic parametrization is simply used in addition, *i.e.* $m := (\ln \epsilon_r, \ln \sigma_r)$, again to avoid negative values of both parameters, but particularly of the conductivity which features values very close to zero. Logarithmic conductivity parametrization also has the advantage to be independent from the arbitrary reference conductivity σ_{ref}. The center frequency $f_c = 100$ [MHz] is used as reference and consequently this reference conductivity is $\sigma_{\text{ref}} \approx 5.6$ [mS/m]. Outer iterations are stopped when satisfying the convergence criterion $J(\ln \epsilon_r, \ln \sigma_r)/J(\ln \epsilon_{r,\text{init}}, \ln \sigma_{r,\text{init}}) < 10^{-3}$. Logarithmic model fields are discretized by hierarchical finite elements of order 1. For this test case, the steepest descent method is no longer considered because it was very slow for the two previous ones and the cross-talk between both parameters is expected to worsen this convergence issue.

Inner product

A cornerstone of the simultaneous reconstruction of both permittivity and conductivity is to favor the update of permittivity during the inversion. Intrinsic sensitivities to both parameters are different and conductivity updates dominate when no appropriate strategy is adopted. In such a situation, conductivity reconstruction features artifacts because it is very sensitive to kinematics, *i.e.* to the permittivity reconstruction. At the opposite, favoring too strongly permittivity reconstruction is also not helpful because high conductivity contrasts are expected to have an important effect on the data [87, 97]. To reach an equilibrium between both updates, several strategies have been proposed. For example, a first idea is to perform two independent line searches in the gradient direction: one for the conductivity part and the other for the permittivity part [112]. The bottleneck of this strategy is that it ignores the possible trade-off between both parameters. Another possibility is then to apply a change of variable on the conductivity, *i.e.* $\sigma_r' = \beta \sigma_r$. This change of variable simply rescales the relative

conductivity such that there is an imbalance in magnitude between the updates of relative permittivity and conductivity. The gradients w.r.t. to the permittivity j'_{ϵ_r} and the rescaled conductivity $j'_{\sigma'_r}$ are actually related to the complex-valued slowness squared gradient j'_{s^2} by the chain rule

$$\begin{aligned}
\{D_{s^2}J\}(\delta s^2) &= \mathrm{Re}\left\langle j'_{s^2}, \delta s^2 \right\rangle && (2.52)\\
&= \mathrm{Re}\left\langle j'_{s^2}, \frac{\partial s^2}{\partial \epsilon_r}\delta\epsilon_r + \frac{\partial s^2}{\partial \sigma'_r}\delta\sigma'_r \right\rangle && (2.53)\\
&= \mathrm{Re}\left\langle j'_{s^2}, \delta\epsilon_r - i\frac{k_0}{k}\frac{1}{\beta}\delta\sigma'_r \right\rangle && (2.54)\\
&= \left\langle \mathrm{Re}\left(j'_{s^2}\right), \delta\epsilon_r \right\rangle + \left\langle -\frac{k_0}{k}\frac{1}{\beta}\,\mathrm{Im}\left(j'_{s^2}\right), \delta\sigma'_r \right\rangle && (2.55)\\
&:= \left\langle j'_{\epsilon_r}, \delta\epsilon_r \right\rangle + \left\langle j'_{\sigma'_r}, \delta\sigma'_r \right\rangle \qquad = \{D_{\epsilon_r,\sigma_r}J\}(\delta\epsilon_r, \delta\sigma'_r). && (2.56)
\end{aligned}$$

Assuming that the real and imaginary parts of the slowness squared gradient have similar amplitudes, the magnitude ratio between the updates of permittivity and unscaled conductivity is then

$$\frac{\left|j'_{\epsilon_r}\right|}{\left|j'_{\sigma'_r}\right|/\beta} \approx \beta^2\frac{k}{k_0}. \tag{2.57}$$

For wavenumbers around the reference k_0 and without rescaling ($\beta = 1$), this ratio is close to one. This is no coincidence: the reference conductivity σ_{ref} is defined to obtain this result. However for values of β bigger than unity, the updates of conductivity are mitigated. This strategy is independent of the optimization algorithm and has thus been used in combination with the *l*-BFGS algorithm, which can advantageously take into account model parameter trade-offs [97]. Here the idea is to introduce this imbalance through a change in the inner product

$$\langle m_a, m_b\rangle_M := \langle \epsilon_{r,a}, \epsilon_{r,b}\rangle + \langle \beta\sigma_{r,a}, \beta\sigma_{r,b}\rangle. \tag{2.58}$$

When β is greater than unity, distances along the conductivity axis are longer than distances along the permittivity axis as

$$\|\Delta\epsilon_r\| = 1 \quad \text{and} \quad \|\Delta\sigma_r\| = 0 \quad \text{yield} \quad \|\Delta m\|_M = 1, \tag{2.59}$$

while

$$\|\Delta\epsilon_r\| = 0 \quad \text{and} \quad \|\Delta\sigma_r\| = 1 \quad \text{yield} \quad \|\Delta m\|_M = \beta. \tag{2.60}$$

Large conductivity updates are therefore further away and are reached later in the inversion process because local optimization methods update their model estimate by local, *i.e.* small, steps. As previously mentioned, changing the inner product is equivalent to preconditioning both the gradient and the Hessian. The approach is thus equivalent to existing strategies

based on preconditioners [87]. The focus of this test case is rather to illustrate how simple inner product preconditioning can handle this balance issue.

Similarly to the previous case studies, the diagonal part of the Gauss-Newton Hessian is used as a weight. Due to the perfect illumination, this first part of this weight can again be approximated by a constant h_{GN} and only the parts related to the logarithmic parametrization are varying spatially

$$\delta s^2 = \frac{\partial s^2}{\partial \ln \epsilon_r} \delta \ln \epsilon_r + \frac{\partial s^2}{\partial \ln \sigma_r} \delta \ln \sigma_r, \tag{2.61}$$

hence

$$w_\epsilon = h_{\mathrm{GN}} \left| \frac{\partial s^2}{\partial \ln \epsilon_r} \right|^2 \quad \text{and} \quad w_\sigma = h_{\mathrm{GN}} \left| \frac{\partial s^2}{\partial \ln \sigma_r} \right|^2 \tag{2.62}$$

and the inner product product is then chosen as

$$\langle m_2, m_1 \rangle_M = \langle \sqrt{w_{\epsilon_r}} \ln \epsilon_{r,2}, \sqrt{w_{\epsilon_r}} \ln \epsilon_{r,1} \rangle + \beta^2 \langle \sqrt{w_{\sigma_r}} \ln \sigma_{r,2}, \sqrt{w_{\sigma_r}} \ln \sigma_{r,1} \rangle . \tag{2.63}$$

According to this choice, the conductivity and permittivity parts of the gradient are given by

$$j'_{\beta, \ln \epsilon_r} = w_{\epsilon_r}^{-1} j'_{\ln \epsilon_r} = w_{\epsilon_r}^{-1} \frac{\partial \epsilon_r}{\partial \ln \epsilon_r} j'_{\epsilon_r} \tag{2.64}$$

and

$$j'_{\beta, \ln \sigma_r} = \beta^{-2} w_{\sigma_r}^{-1} j'_{\ln \sigma_r} = \beta^{-2} w_{\sigma_r}^{-1} \frac{\partial \sigma_r}{\partial \ln \sigma_r} j'_{\sigma_r}. \tag{2.65}$$

The magnitude ratio between the updates of permittivity and conductivity is then

$$\begin{aligned}
\frac{\left| \epsilon_r \left(\exp j'_{\beta, \ln \epsilon_r} - 1 \right) \right|}{\left| \sigma_r \left(\exp j'_{\beta, \ln \sigma_r} - 1 \right) \right|} \approx \frac{\epsilon_r \left| j'_{\beta, \ln \epsilon_r} \right|}{\sigma_r \left| j'_{\beta, \ln \sigma_r} \right|} &= \frac{\epsilon_r \left| w_{\epsilon_r}^{-1} \frac{\partial \epsilon_r}{\partial \ln \epsilon_r} j'_{\epsilon_r} \right|}{\sigma_r \left| \beta^{-2} w_{\sigma_r}^{-1} \frac{\partial \sigma_r}{\partial \ln \sigma_r} j'_{\sigma_r} \right|} \\
&= \frac{\left| \frac{\partial s^2}{\partial \ln \epsilon_r} \right|^{-2} \left| \frac{\partial \epsilon_r}{\partial \ln \epsilon_r} \right|^2}{\left| \frac{\partial s^2}{\partial \ln \sigma_r} \right|^{-2} \left| \frac{\partial \sigma_r}{\partial \ln \sigma_r} \right|^2} \beta^2 \frac{\left| j'_{\epsilon_r} \right|}{\left| j'_{\sigma_r} \right|} \\
&= \frac{\left| \frac{\partial s^2}{\partial \epsilon_r} \right|^{-2}}{\left| \frac{\partial s^2}{\partial \sigma_r} \right|^{-2}} \beta^2 \frac{\left| j'_{\epsilon_r} \right|}{\left| j'_{\sigma_r} \right|} \\
&= \frac{k_0^2}{k^2} \beta^2 \frac{k}{k_0} \\
&= \beta^2 \frac{k_0}{k}.
\end{aligned}$$

Again, for wavenumbers around the reference k_0, this ratio is close to β^2. However, the behaviour around the reference is reversed: higher wavenumbers give more weight to the conductivity part. It is interesting to highlight that this comes from the diagonal Hessian approximation and not from logarithmic parametrization.

The line search *l*-BFGS algorithm has been applied for three values of the scaling coefficient β.

Convergence curves are given in Figure 2.18 while inversion results are given in Figure 2.19. As far as the permittivity is concerned, all values of the scaling parameter give similar reconstructions. Conductivity reconstructions are however more sensitive to the scaling parameter and artifacts appears when the scaling parameters is too low, *e.g.* when $\beta = 2$. For smaller values, the algorithm can even fail to converge. An analysis of the evolution of the reconstruction over the iterations actually shows (as expected) that high scaling parameters β delays conductivity updates, hence preventing harmful artifacts from appearing (*cf.* Appendix D). When convergence can be reached, the bigger β is, the (slightly) smoother the conductivity reconstruction is. Unfortunately, the convergence also slows down when the scaling parameter increases. Consequently for the sequel of this case study, the smallest scaling parameter that gives an acceptable conductivity reconstruction is chosen, *i.e.* $\beta = 3$. The performance of the *l*-BFGS and the Newton optimization methods is described in the next two subsections. Convergence curves, inversion results and statistics are given in Figure 2.20 and 2.21 and in Table 2.6 respectively.

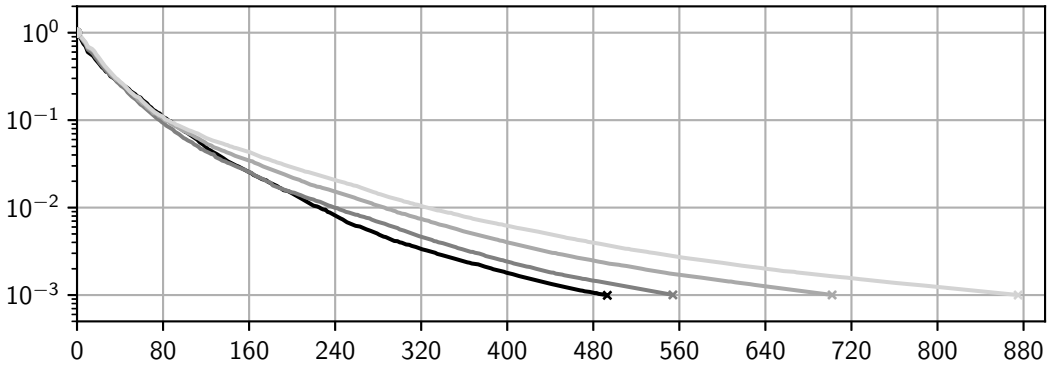


Figure 2.18: Data misfit as a function of the computational complexity for the line search *l*-BFGS algorithm with different scaling for the inner product inner product. $\beta = 2$ (▬), $\beta = 3$ (▬), $\beta = 4$ (▬), $\beta = 5$ (▬).

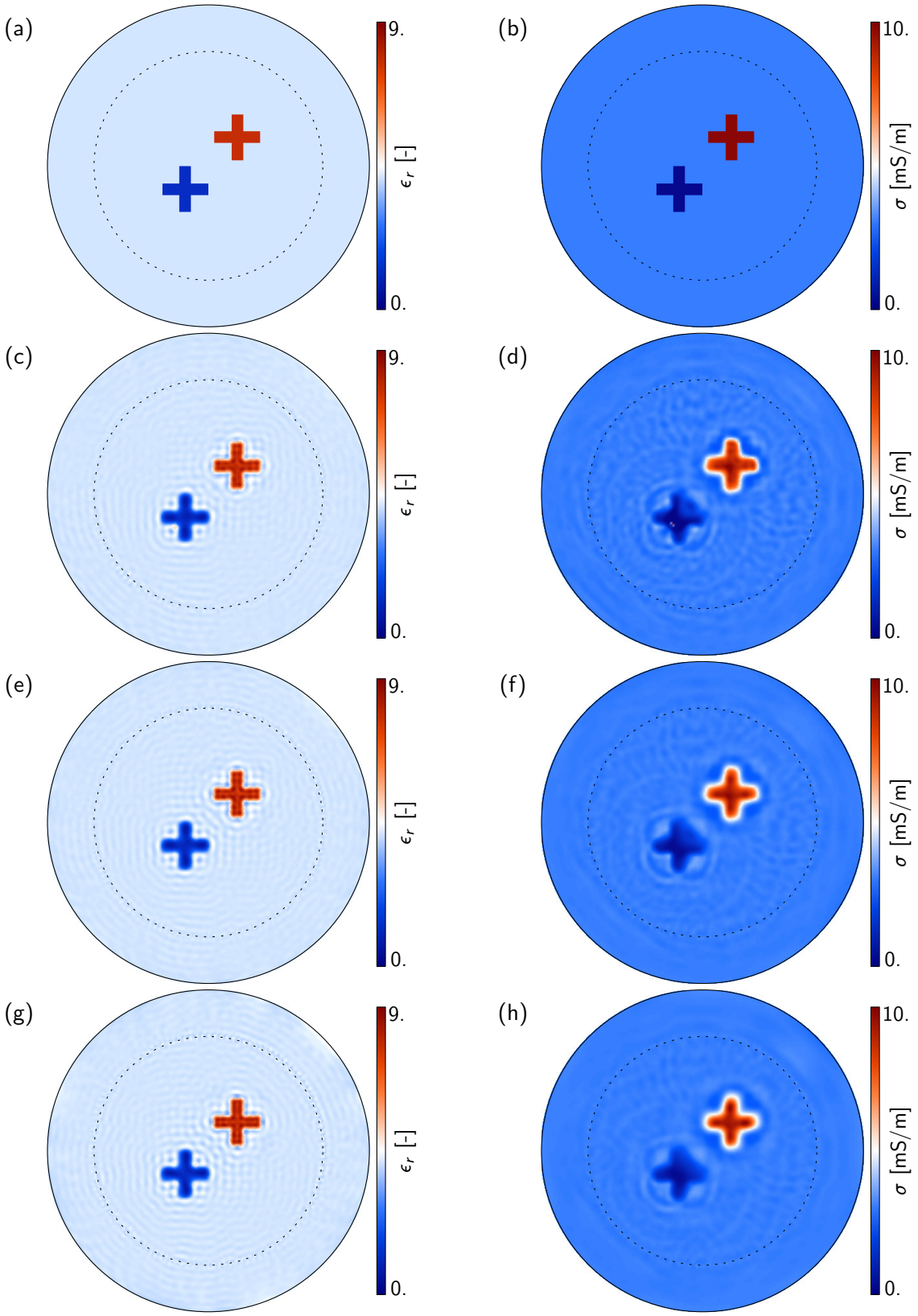


Figure 2.19: Final inversion results for the line search *l*-BFGS algorithm with different scaling for the inner product. $\beta = 2$ (c,d), $\beta = 3$ (e,f), $\beta = 5$ (g,h). The first row (a,b) is the true model. Left and right columns are respectively permittivity and conductivity reconstructions.

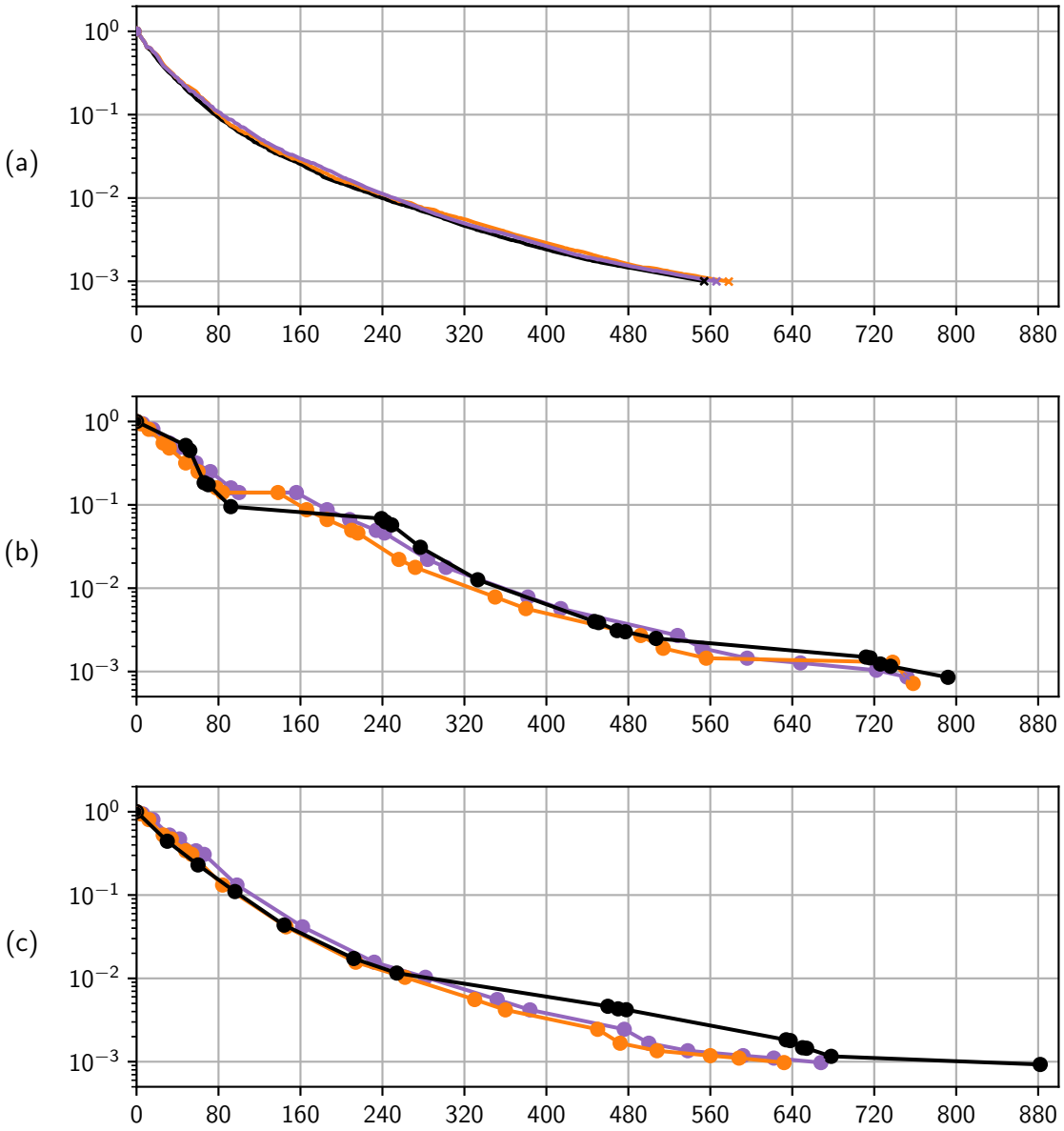


Figure 2.20: Data misfit as a function of the computational complexity for the *l*-BFGS (a), the full Newton (b) and the Gauss-Newton (c) methods combined with either a line search (▬) or a prospective trust region (B (▬)) or a retrospective trust region (B (▬)). Dots on (Gauss)-Newton curves indicate outer iterations.

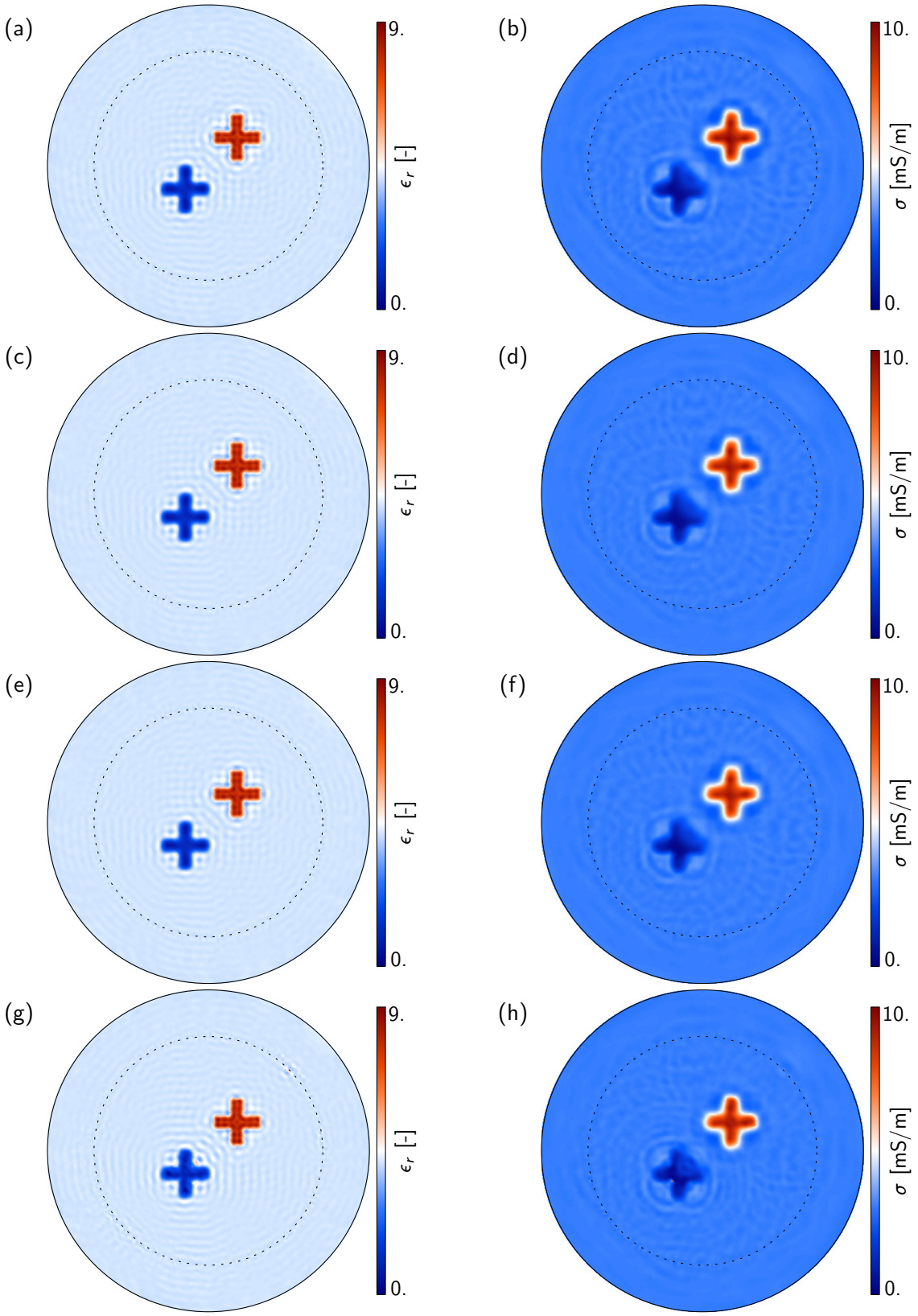


Figure 2.21: Inversion results for the l-BFGS (a,b), the full Newton (c,d) and the Gauss-Newton (e,f) methods combined with trust-region method using a retrospective radius update (B) and for the full Newton method combined with a line search (g,h).

		Wave sol. (tot)	Outer it. (tot)	Inner it. (avg)	Rejected (%)	Constrained (%)	Negative curv. (%)
LB	LS	554	272	-	4	-	-
	TR-P (B)	578	289	-	6	28	-
	TR-R (B)	566	283	-	4	5	-
FN	LS	792	20	18.45	15	-	10
	TR-P (B)	758	22	16.2	5	55	0
	TR-R (B)	752	23	14.3	4	57	0
GN	LS	882	15	28.3	7	-	-
	TR-P (B)	632	18	16.6	0	44	-
	TR-R (B)	668	18	16.6	0	44	-

Table 2.6: Statistics related to the implementation of the *l*-BFGS (LB), the full Newton (FN) and the Gauss-newton (GN) methods combined with a line search (LS) or a trust-region (TR) with a prospective (P) or retrospective (R) radius update with parameter set B.

Limited memory BFGS method

The influence of the globalization method on the convergence speed is again small, particularly for the retrospective trust-region method and the line search method which barely accept the pure *l*-BFGS direction at each iteration. Now the only difference with the prospective trust-region method is that the constraint is active more often, although it does not slows down the convergence dramatically.

Newton methods

Inversion results for all the methods are similar, except for the Newton method combined with a line search (*cf.* Figure 2.21g and 2.21h) whose conductivity reconstruction of the lower cross is less accurate. Again, the full Newton and the Gauss-Newton methods combined with a line search struggle to end the iterative solver at the right time. Indeed, the full Newton method is stopped twice because of negative curvature while the Gauss-newton method is stopped twice before convergence but after 100 iterations. From the history of inner iterations per outer iterations (*cf.* Figure 2.22) and the history of forcing sequence (*cf.* Figure 2.23), it can be seen that rebounds in the forcing sequence occur at the same time than inner iteration peaks. Again, it means that a lot of computational effort, *i.e.* a lot of inner iterations, is incurred for a system which is seen *a posteriori* to be inaccurate, as shown by the rebounds in

the forcing sequence. This over-solving decreases the convergence speed. For this case study, the burst flow of line search methods, *i.e.* an accurate system resolution followed several inaccurate resolution, seems to be less appropriate than a more steady system resolution strategy, as provided by trust-region methods. As far as trust-regions are concerned, the best combination in this case is the Gauss-Newton method with a prospective radius update (B). The corresponding retrospective method give the exact same solution because the conclusions from both the prospective and retrospective ratios are always the same. The latter only takes two supplementary wave solutions per outer iteration for the computations of the retrospective ratio, *i.e.* 36 more waves solutions. Similarly to the first test case, the prospective ratio is larger than unity for the first few outer iterations while the retrospective ratio is smaller than unity (*cf.* Figure 2.24). However, the discrepancy away from unity is smaller than for the first two test cases, probably because multiple scattering is weaker. There is thus no early over-solving that could be corrected by the retrospective radius update, or by the full Newton Hessian. For that reason and because it rejects one long iteration, the Newton method is not faster in this case. Retrospective and prospective radius update converge at the same speed, because the conclusions at a single late iteration differ (at the 18^{th} iteration, $\rho_{\text{p}} = 0.78$ while $\rho_{\text{r}} = 0.69$). Retrospective radius update can quickly compensate its cost of two additional wave solutions when the number of inner iterations per outer iteration is high and hence it might become competitive.

Interestingly, for the first two case studies the *l*-BFGS method clearly outperforms any Newton method: it is nearly twice faster. For this third test case, convergence speed are comparable, even if the *l*-BFGS method is still slightly faster. This observation confirms that Newton methods might be interesting for multi-parameter inversions [117].

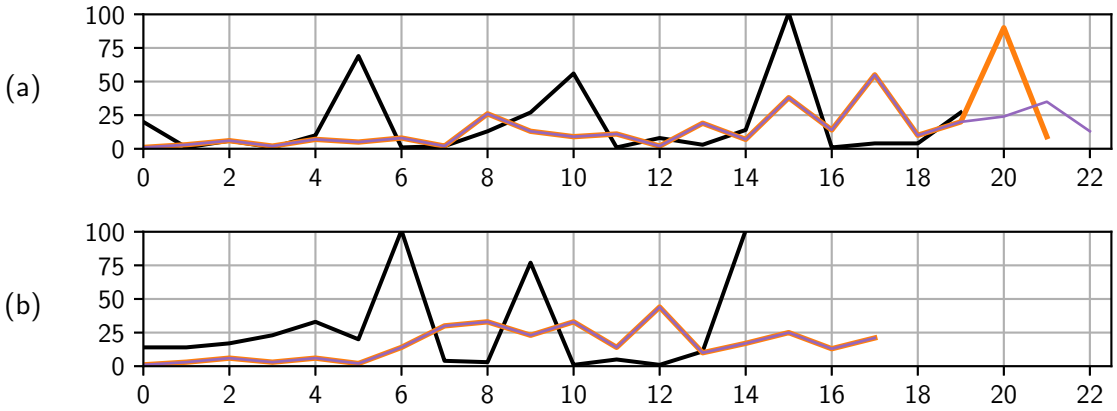


Figure 2.22: Inner iterations per outer iteration for the full Newton method (a) and the Gauss-Newton method (b) combined with either a line search (▬) or a trust-region with a prospective (B (▬)) or a retrospective (B (▬)) radius update.

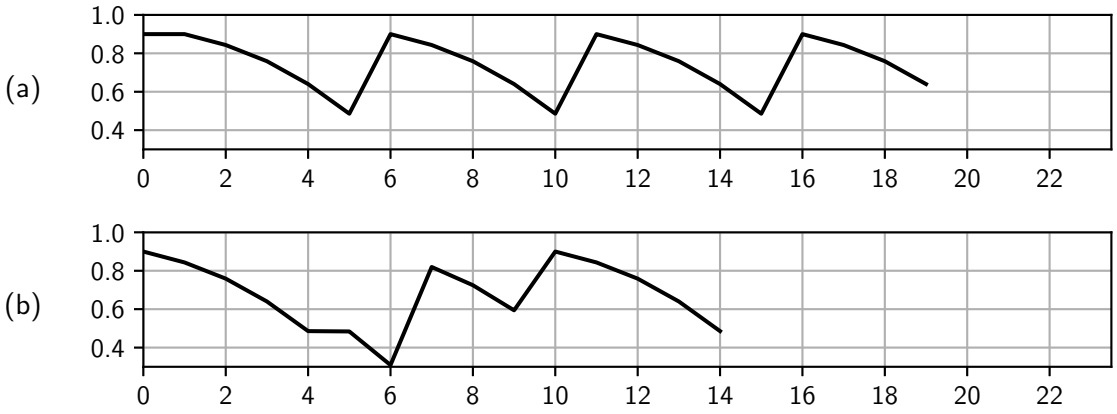


Figure 2.23: Forcing sequence η for the full Newton (a) and the Gauss-Newton (b) methods combined with a line search method (▬). The forcing sequence for methods combined with a trust-region is constant ($\eta = 0.5$).

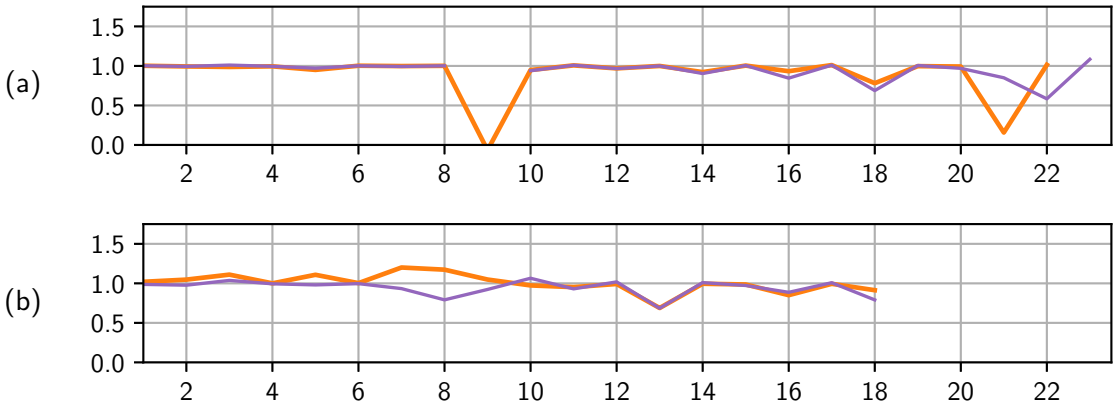


Figure 2.24: Prospective ratio ρ_p using a prospective radius update (B (▬)) or retrospective ratio ρ_r using a retrospective radius update (B (▬)) during the outer iterations of the full Newton method (a) and the Gauss-Newton method (b).

2.4 Conclusion

In this chapter, several local optimization algorithms, based on combinations of descent directions, globalization strategies and inner product modifications, have been investigated in the context full waveform inversion. The three most widely used descent directions have been considered, in combination with scaling, weighting and/or smoothing inner products and with line search or trust-region globalization strategies. At the heart of any trust-region method is the trust-region constraint, which is expressed in terms of the inner product chosen for the model parameter space. Consequently, the analysis began by investigating different inner product choices that could be implemented. It was shown that changing the inner product does not only modify how lengths are measured but also acts as a preconditioner on both the gradient and the Hessian operator. Based on the three reference numerical case studies, it was shown that moving from the conventional inner product to a smoothed and/or weighted inner product can accelerate the convergence and mitigate the non-linearity of the misfit, for any optimization method independently of the globalization method (line search or trust-region). In parallel with this inner product choice, line search and trust-region variants of the steepest descent, the *l*-BFGS and the (Gauss-)Newton methods, were introduced. The number of wave propagation problems to be solved for each method was derived in order to compare them fairly. For each optimization method, the line search and the trust-region globalizations were then compared based on the different case studies. Thanks to the inner product preconditioning, every combination actually already yielded very satisfying results. Nevertheless, trust-region methods outperform line search methods in numerous situations. In particular, the steepest descent converges slightly faster, because the trust-region methods always tried to increase the step length. As far as the *l*-BFGS method is concerned, very few differences were noted, but interestingly, constraining the size of the update direction did not decrease the convergence speed. The more dramatic differences appeared when using Newton methods. Trust-region methods actually overcome the difficulties, related to the small or negative eigenvalues of the (Gauss-)Newton Hessian operator, that appeared when using a line search method. The optimization flow related to both globalizations method was also very different: line search methods alternated very few and many inner iterations per outer iterations while trust region methods were more steady. This latter strategy has shown to be more efficient, both in terms of convergence speed and in terms of reconstruction accuracy. When multiple scattering is strong, the Gauss-Newton approximation becomes less accurate, and consequently the associated misfit predictions, involved in trust-region methods, are also less accurate. Typically, the prospective ratio was bigger than one while the retrospective ratio was smaller than one and it was then interesting to use the latter to avoid over-solving, even if its computational cost is high. Actually a better solution was to use the full Newton Hessian, and to get rid of the approximation. The full Newton method keeps an accurate misfit

representation, even when multiple scattering is strong and hence converge faster. Moreover, when multiple scattering was weak, the converge speed of the full Newton method was still comparable to the Gauss-newton method. Because of its good performance independently of the configuration, the full Newton method combine with a trust-region globalization method is really good choice, as the *l*-BFGS method.

More sophisticated optimization methods, for example combining *l*-BFGS and Newton methods or trust-region and line search methods, could increase even more the convergence speed. Based on this chapter and its potential extensions, trust-region methods and inner product preconditioning seem to be two very useful tools for full waveform inversion.

In this chapter, all the numerical case studies were noiseless. Noiseless inversion is an interesting first step because it is simpler but still allow to highlight some issues related to full waveform inversion. The next step towards real inversions is to use more realistic data, typically contaminated by noise. This is the focus of the following chapter, in which the behavior of inner product preconditioned methods in the presence of noise is investigated. In particular, the *per se* regularization effect of the inner product, due to its preconditioning properties, is investigated. A more sophisticated inner product that involves prior information on the model parameter space is also introduced to illustrate further the possibilities given by inner product modifications.

Chapter 3

Regularization strategies

Full waveform inversion is an imaging method that aims to minimize the discrepancy between some measured data and their numerically simulated counterpart. When these measured data are noisy, the smallest discrepancy does not necessarily coincide with the best solution, because a perfect match actually means that the simulated data partially matches noise. In order to avoid such over-fitting, multiple strategies exist. The most widely used technique consists in adding a penalization term to the data misfit to be minimized, while the simplest strategy consists in stopping the iterative minimization algorithm before it reaches the actual minimum. This latter option is however not very efficient unless it is combined with preconditioning, for example through inner product modifications, because preconditioning allows to delay the appearance of noise artifacts. This second strategy is the focus of this chapter. In particular, two inner product preconditioning strategies are considered. Both are based on Laplace filtering: the first one being isotropic and the second one anisotropic. The second inner product is slightly more elaborate and is strongly inspired by the so-called *edge-enhancing operator* from the image diffusion filtering community. This operator is adaptive as it evolves during the inversion process to stop smoothing across sharp interfaces, at the opposite of the stationary isotropic variant. Sharp interfaces are therefore expected to be reconstructed faster and more accurately. Based on the three reference numerical case studies, this novel regularization strategy is compared to more conventional additive penalization strategy. In particular, it is shown that the early stopping method with preconditioning yields a higher reconstruction quality than the penalization method in numerous situations, while featuring desirable advantages: firstly it is more flexible because the inner product can be modified at any outer iteration; then it is more robust because the regularization is active from the very beginning of the inversion thanks to preconditioning; and finally it is easier to choose the regularization hyper-parameters because they can be interpreted as smoothing lengths.

Highlights

- Comparative review of regularization methods: inner product preconditioning versus additive misfit penalization
- Introduction of the edge-enhancing inner product
- Illustration on the three reference numerical case studies

3.1 Introduction

Full waveform inversion is a data-fitting procedure used to reconstruct some high-resolution quantitative model parameters m by minimizing the distance between some observed data d_0 and the corresponding simulated data $d(m)$, obtained though a forward model $F(m)$ and a measurement operator R

$$m^* = \arg\min J_d(m) \quad \text{with}$$
$$J_d(m) := \text{dist}\left(d(m), d_0\right), \quad d = R(u) \quad \text{and} \quad F(m)u = f. \quad (3.1)$$

The information extracted from the wave field u is however very limited, as R is typically a projection operator on a small number of punctual receivers at a few frequencies. Hence there might exist multiple equivalent minimizer m^*. Moreover, in a realistic setting, multiple error sources exist in the workflow. For example, the direct model $F(m)$ is never exact, which implies that the ground-truth model parameters $\tilde{m}$ does not yield a perfect match, *i.e.* $d(\tilde{m}) \neq d_0$, and real measured data are always contaminated by noise. For both these reasons, the realistic input of an inverse problem is a noisy dataset $d_{0,n}$, rather than the perfect data set d_0. This noisy data set encapsulates both direct operator errors and noise. A straightforward minimization as stated by (3.1) drives the model parameters towards a minimizer such that $d(m) \to d_{0,n}$ while of course the real target is a minimizer such that $d(m) \to d_0$.
Solution procedures based on (3.1) potentially have multiple solutions and these solutions strongly depend on the noise in the data. Hence solving the inverse problem through (3.1) is not appropriate in the presence of noise and the procedure should be modified to tackle these difficulties, that are due to the inherently ill-posedness of the inverse problem. Such strategies are often called *regularization method* [44, 86]. Basically, these regularization techniques aim to furnish prior information about model parameters to the inversion, in order to discriminate between multiple possible solutions, but also to prevent the simulated data from over-fitting the noise in the measured data. The choice among different regularization techniques thus depends on the specific problem, and it can be governed by the need to preserve or emphasize particular features of the model parameters, *e.g.* piece-wise smooth distributions as far as

natural media are concerned.

There are several ways to incorporate this prior information into the optimization problem, that can be sorted in two categories: *explicit regularization*, where the model parameters are expanded in an appropriate basis that contains only the desirable features [66, 67, 83] and *implicit regularization*, where undesirable features are penalized but can still appear if they explain the data very well. The most common implicit regularization method is probably to add a penalization term to the data misfit [11]

$$J(m) = J_d(m) + \lambda J_m(m). \tag{3.2}$$

Ideally, this additive penalization term should favor the most likely model parameter distributions, *i.e.* it should be small for piece-wise smooth distributions. Tikhonov penalization of the model parameter gradient is particularly well suited for retriving smooth model parameters [181]

$$J_m(m) := \|\mathbf{grad}\,(m)\|_{L_2}^2 = \int_\Omega |\mathbf{grad}\,(m)|^2 \; d\Omega. \tag{3.3}$$

However for models with sharp edges, total variation penalization is more appropriate [159]

$$J_m(m) := \|\mathbf{grad}\,(m)\|_{L_1} = \int_\Omega |\mathbf{grad}\,(m)| \; d\Omega. \tag{3.4}$$

Several recent regularization methods are based on variants of the total variation penalization [8]. For example, a blocky penalization term can be used instead [69]. This term identifies discontinuities in the model parameters and penalizes derivatives along these discontinuities, which facilitates the automatic delineation of high-contrast model parameters. Similarly, the total variation functional can be modified to feature a vertically asymmetric part that only penalizes the depth-deacreasing model discontinuities, which also helps the apparition of large and deep highly-contrasted strucutres [49, 84]. The penalization (3.3) can also be modified to be anisotropic thanks to some tensor $\boldsymbol{D}$, which is typically adaptative, in the sense that it follows the structures of the current model parameters [14, 65]

$$J_m(m) := \int_\Omega |\boldsymbol{D} \cdot \mathbf{grad}\,(m)|^2 \; d\Omega. \tag{3.5}$$

Interestingly, any such model functional can also be used as a strong bound constraint for the minization problem, rather than a weak misfit penalization [49, 140]. More sophisticated constrained optimization algorithm must however be used in that case. Both Tihkonov and total variation penalization terms are respectively defined as the (squared) L_2 norm and the L_1 norm of the model spatial derivatives, as the L_p norm is defined as

$$\|\mathbf{grad}\,(m)\|_{L_p} = \left(\int_\Omega |\mathbf{grad}\,(m)|^p \; d\Omega \right)^{1/p}. \tag{3.6}$$

One reason for the success of regularization strategies based on total variation is the increased sparsity of model spatial derivatives, obtained thanks to the decrease from $p = 2$ to $p = 1$. To increase this sparsity even further, non-convex penalization terms, based on the L_p norm with $0 < p < 1$, can be used [59]. Unfortunately, non-convexity, which is at the root of these methods, also complicates the optimization procedure. Instead of building new functionals, another approach consists in interlacing the Tikhonov regularization term with an image denoising problem. This image denoising problem is based on a first or second order total variation problem. Combining both functionals allows to reconstruct both sharp interfaces and smooth variations at the expense of solving two coupled optimization problems, while high-order derivatives are rather used to avoid the typical staircase effect from the first-order total variation [41, 59, 103]. Data misfit minimization can also be interlaced with a denoising problem through proximal Newton methods, which generalize the traditional Newton step to involve the sub-gradients of a possibly non-differentiable regularization function through operator splitting and proximal mappings [4].

All the above mentioned regularization techniques are based on model spatial derivatives and can thus be seen as low-pass filters from the Fourier domain perspective. Denoting the spatial frequency by $\boldsymbol{k}$ and the spatial Fourier transform of the model parameters by $\hat{m}$, the simplest example is the Tikhonov penalization

$$\|\mathbf{grad}\,(m)\|_{L_2}^2 = \int_{\mathbb{R}^2} \boldsymbol{k}^2 \, |\hat{m}(\boldsymbol{k})|^2 \; d\boldsymbol{k}. \tag{3.7}$$

It immediately appears that it is equivalent to penalize the high-frequency spatial oscillation during the reconstruction. However, model parameters are not necessarily sparse in the Fourier domain, meaning that the ground-truth model parameters have a lot of non-vanishing high-frequency Fourier components. Following this observation, it seems natural to penalize the coefficients of an appropriate transform, specifically one in which the ground-truth model parameters are expected to be sparse. Decomposition on a wavelet [107] or a seislet [199] basis have proven to be particularly successful. An adaptive transform, whose dictionary is learned from the descent directions of previous iterations has also been proposed recently [209]. The sparsity in the transform domain is again imposed using a L_1 or L_0 penalization or constraint and can lead to some modifications of the optimization algorithm.

A major drawback of additive penalization comes from the compromise between the data and the model term. Indeed, the data term J_d tends to add details in the image while the regularization term tends to smooth them out. The balance between these two objective is set through the regularization parameter λ. A large part of the data must be explained by the model parameters and consequently the regularization parameter is chosen relatively small. If not, then the model parameters are mainly driven by prior information. In the former case, the influence of the regularization term only becomes visible at the end of the inversion process such that during the earliest iterations, noise can freely deteriorate the reconstruction.

In order to avoid noise artifacts from the beginning of the inversion but also to get rid of the regularization parameter λ (which can be hard and expensive to select), multiplicative regularization has been introduced [12]. Multiplicative regularization proposes to multiply, instead of adding, the data misfit and the regularization term, *i.e.*

$$J(m) = J_d(m) \times J_m(m). \tag{3.8}$$

There is thus no regularization parameter and the influence of the regularization term is present from the beginning to the end of the inversion. In addition to removing noise artifacts, early regularization could also increase the robustness in terms of starting models [79, 152]. The flip side of the performance functional (3.8) is a possible increase of the non-linearity of the inversion problem.

All the methods briefly described here above provide very successful results. However, they often require more advanced local optimization methods, because of either a strong bound constraint [49, 107, 209], a non-convex L_p norm term [59], two interlaced optimization problems [59, 103] or a slightly more non-linear performance functional [12, 79, 152]. In addition, any regularization strategy based on an additive misfit penalization fails to constrain the model reconstruction from the beginning of the inversion and is basically based on a compromise. Regularization through preconditioning has been introduced for these specific two reasons. The root of this last family of methods is to apply a preconditioner to the descent direction at each outer iteration. Hence it can be combined with any existing local optimization algorithm without modifications and its effect appears during the whole inversion. The most straightforward preconditioning approach is to smooth the descent direction with the same strength in all directions, *e.g.* with an isotropic Laplcian filter. However, it is much more interesting to include prior knowledge in this preconditioner, for example to prevent smoothing to occur along directions that feature sharp interfaces. Such a smoothing can easily be done through directional filters and anisotropic Laplacian operators [58, 70, 73, 99, 196, 201, 210]. More recently, a diffusion filtering operator, *i.e.* a non-linear operator, has also been applied successfully to the descent direction [114, 204]. When using preconditioning as a regularization strategy, the misfit is composed exclusively of the data part: there is no compromise in the misfit. The iterative minimization process should however be stopped on time, before noise inevitably introduces artifacts in the reconstruction. Regularization techniques developed in this chapter fall in this last category; their particularity being that preconditioning is here brought through inner product modifications, which define the associated preconditioner.

This chapter is organized in two parts as follows. In Section 3.2, two widely used additive misfit regularization strategies are quickly reviewed. Specifically, Tikhonov and total variation penalization terms are considered. Regularization through inner product modifications is then introduced. The two proposed inner products are derived starting from the two above-mentioned penalization terms, in order to highlight the connections between both approaches.

In particular, the link between a misfit penalty and an inner product penalty is established. In Section 3.3, both family of methods are applied to the three reference numerical case studies. Each of them highlights some advantages of regularization through preconditioning. The first case study aims to reconstruct the Marmousi model, which features numerous linear structures. Hence the edge-enhancing inner product yields faster convergence and more accurate reconstruction than the isotropic variant. For the second case study, both conventional additive misfit penalization fail to reconstruct accurate models, because regularization is not enforced from the beginning of the inversion, while preconditioned based method do succeed. Finally, the third case study shows a glimpse of the new possibilities and the flexibility offered by inner product preconditioning. Specifically, it is here used to introduce a soft coupling between the two parameter classes, because conductivity and permittivity anomalies are typically correlated but the conductivity is often less constrained by the data. Throughout all the case studies, the ease to select the inner product hyper-parameters is also emphasized, in particular in comparison with the selection of the regularization parameter λ, which can be hard and expensive to compute. Moreover, inner product preconditioning is also shown to be appropriate for inversion with limited computational resources, where the optimization must be stopped before the optimal point, again because the regularization acts on the whole optimization path.

3.2 Regularization strategies

3.2.1 Additive penalization

As mentioned in the introduction, in additive penalization methods, the misfit is augmented with a penalization term $J_m(m)$. This penalization term also adds a contribution to the first and second order directional derivatives, hence also to the gradient and to the Hessian operator. The particular expression of course depends on the particular regularization that is considered. Analyzing these contributions can help understand why Tikhonov and total variation regularizations (3.3)-(3.4) are dramatically different while both are base on a L_p norm.

Tikhonov penalization The first and second order directional derivatives of the Tikhonov penalization (3.3) are straightforward to compute

$$\{D_m J_m(m)\}(\delta m) = \int_\Omega \mathbf{grad}(m) \cdot \mathbf{grad}(\delta m)\, d\Omega \tag{3.9}$$

and

$$\left\{D^2_{mm} J_m(m)\right\}(\delta m_1, \delta m_2) = \int_\Omega \mathbf{grad}(\delta m_2) \cdot \mathbf{grad}(\delta m_1)\, d\Omega. \tag{3.10}$$

Neglecting the boundary term, the conventional gradient is then simply

$$j'_m = -\mathrm{div}\left(\mathbf{grad}\left(m\right)\right). \tag{3.11}$$

To understand the meaning of this gradient, considering an hypothetical line search steepest descent algorithm is helpful. In such a setting, the evolution of the model parameters is

$$\Delta m = -\gamma\left(j'_m + j'_d\right), \tag{3.12}$$

where j'_d is the gradient related to the data misfit J_d. If infinitesimal steps are taken, *i.e.* $\gamma := \Delta\tau \to 0$, this evolution can be written as a diffusion process

$$\frac{\partial m}{\partial \tau} = -(j'_m + j'_d). \tag{3.13}$$

Dropping the data term, the model parameter evolution is therefore a simple isotropic diffusion, *i.e.* $\frac{\partial m}{\partial \tau} = -\mathrm{div}\left(\mathbf{grad}\left(m\right)\right)$, whose steady state solution is a constant. As mentioned previously, the Tikhonov penalization therefore drives to smooth any spatial variation of the model parameters. This smoothing effect is illustrated schematically in Figure 3.1 for one dimensional edges and peaks. Intuition about the evolution of both structures is easily obtained when looking at the sign of derivatives.

Total variation penalization Computing the derivatives of the total variation penalization is less straightforward (*cf.* Appendix E). After some manipulations, the first order derivative can be written

$$\left\{D_m J_m(m)\right\}(\delta m) = \int_\Omega \frac{\mathbf{grad}\left(m\right)}{\left|\mathbf{grad}\left(m\right)\right|} \cdot \mathbf{grad}\left(\delta m\right)\, d\Omega. \tag{3.14}$$

The second order derivative can be expressed compactly in terms of the diffusion amplitude $g_\perp$ and the perpendicular and parallel structure tensors $\boldsymbol{D}_\perp$ and $\boldsymbol{D}_\parallel$

$$\left\{D^2_{mm} J_m(m)\right\}(\delta m_1, \delta m_2) = \int_\Omega g_\perp(\left|\mathbf{grad}\left(m\right)\right|^2)\ \mathbf{grad}\left(\delta m_2\right) \cdot \boldsymbol{D}_\perp \cdot \mathbf{grad}\left(\delta m_1\right)\, d\Omega \tag{3.15}$$

with

$$g_\perp(r^2) = \frac{1}{r}, \qquad \boldsymbol{D}_\perp = \boldsymbol{I} - \boldsymbol{D}_\parallel \quad \text{and} \quad \boldsymbol{D}_\parallel := \frac{\mathbf{grad}\left(m\right) \otimes \mathbf{grad}\left(m\right)}{\left|\mathbf{grad}\left(m\right)\right|^2}. \tag{3.16}$$

These notations will be particularly useful when building the edge-enhancing inner product in the next subsection. Note that at the origin, when $\left|\mathbf{grad}\left(m\right)\right| = 0$, the total variation penalization is not differentiable and a small threshold β must be added to prevent numerical instabilities, *i.e.* $\left|\mathbf{grad}\left(m\right)\right| \to \sqrt{\left|\mathbf{grad}\left(m\right)\right|^2 + \beta^2}$. The first directional derivative is very different from the previous case. Consequently, the conventional gradient is also very different

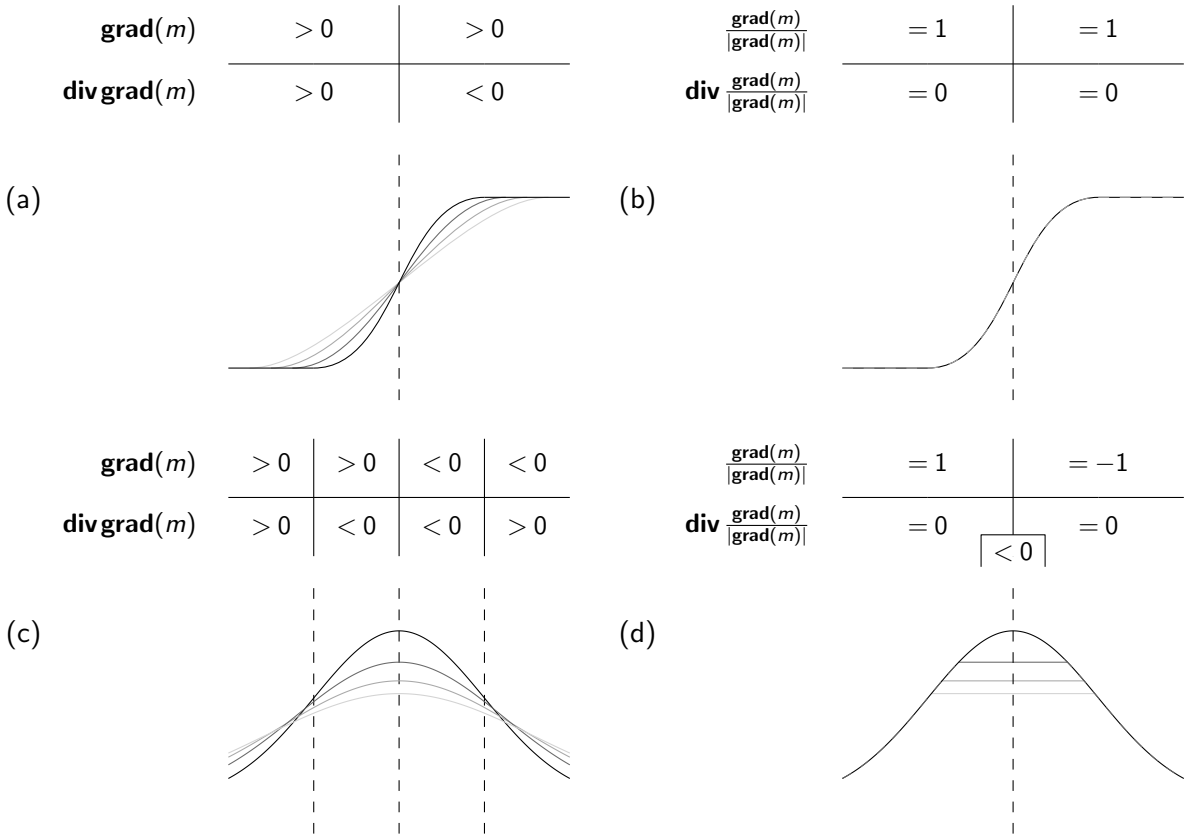


Figure 3.1: Schematic illustration of the evolution of edges (a,b) and peaks (c,d) under Tikhonov (a,c) and total variation (b,d) regularization strategies. The initial model parameters is the error function for edges, *i.e.* $m_0(x) = \mathrm{erf}(x)$, and the Gaussian function for peaks, *i.e.* $m_0(x) = \exp(-x^2)$. Model parameters are shown at different pseudo-time τ_0 (▬), τ_1 (▬), τ_2 (▬), τ_3 (▬) ($0 = \tau_0 < \tau_1 < \tau_2 < \tau_3$).

and the associated diffusion (3.13) does not evolve in the same direction

$$j'_m = -\mathrm{div}\left(\frac{\mathbf{grad}\,(m)}{|\mathbf{grad}\,(m)|}\right). \tag{3.17}$$

Indeed, this gradient cancels everywhere except when it switches sign, *i.e.* on peaks: maximum or minimum. At the opposite of the Tikhonov regularization, edges are thus preserved, because the spatial derivatives is either always positive or always negative and peaks tend to turn into plateaus. This evolution is again illustrated in Figure 3.1, from a sign analysis of the gradient. Details on Tikhonov and total variation regularization and in particular their link with diffusion filtering can be found in [162].

3.2.2 Inner product preconditioning

The first effect of changing the inner product is to modify both the gradient and the Hessian operator

$$\langle m_2, m_1 \rangle_M = \langle P m_2, m_1 \rangle \quad \Rightarrow \quad j'_{d,M} = P^{-1} j'_d \text{ and } H_{d,M} = P^{-1} H_d. \tag{3.18}$$

The model parameter evolution can again be written as a diffusion process, which now depends on the preconditioner P

$$\frac{\partial m}{\partial \tau} = -P^{-1} j'_d. \tag{3.19}$$

Interestingly, when the data misfit gradient vanishes, *i.e.* $j'_d = 0$, the process stops. Stated otherwise, if the initial model m_0 is a local minima of the data misfit, *i.e.* $j'_d(m_0) = 0$, then the model parameters will not be updated. This behavior is actually very different from additive penalization, which seeks a compromise between the data misfit and the penalization. Preconditioning exclusively acts on the optimization path to the minimizer, but not on the minimizer itself. Nevertheless, in the case the data misfit features multiple local minima, changing the path can change the particular minimizer that is reached and in the case the data misfit features a wide global minimum plateau, it can then modify the approximate minimizer that can be reached at a limited computational cost. The inner product should then be designed such that the application of the associated preconditioner favors desirable features. For the remainder of this section, the subscript 'd' is omitted for compactness, as there is no other misfit anymore.

Preconditioning is here introduced through the inner product, meaning that the operator P is a consequence of the inner product choice $\langle \cdot, \cdot \rangle_M$. Instead of penalizing the misfit, the idea is to penalize the inner product

$$\langle \delta m_2, \delta m_1 \rangle_M = \int_\Omega \delta m_2 \, \delta m_1 \, d\Omega + \int_\Omega h(\delta m_2, \delta m_1) \, d\Omega \quad \text{such that} \quad \|\delta m\|_M \geq \|\delta m\|_{L_2}. \tag{3.20}$$

A semi-positive definite bi-linear approximation of some additive misfit penalization is typically used as a starting point for this inner product penalty

$$\int_\Omega h(\delta m_2, \delta m_1) \, d\Omega = \left\{ D^2_{mm} J_m(m) \right\} (\delta m_1, \delta m_2). \tag{3.21}$$

Whatever the inner product choice is, the decrease in the conventional gradient direction remains the same, *i.e.* $\{D_m J_d(m)\}(j'_{L_2}) = \left\| j'_{L_2} \right\|^2_{L_2}$, and it is always larger than the decrease in the preconditioned gradient direction, *i.e.* $\{D_m J_d(m)\}(j'_M) = \|j'_M\|^2_M$:

$$\{D_m J_d(m)\} \left(\frac{j'_M}{\|j'_M\|_M} \right) = \|j'_M\|_M \tag{3.22}$$

$$= \frac{\left\langle j'_{L_2}, j'_M \right\rangle_{L_2}}{\|j'_M\|_M} \tag{3.23}$$

$$\leq \frac{\left\|j'_{L_2}\right\|_{L_2} \|j'_M\|_{L_2}}{\|j'_M\|_M} \leq \left\|j'_{L_2}\right\|_{L_2} = \{D_m J_d(m)\} \left(\frac{j'_{L_2}}{\left\|j'_{L_2}\right\|_{L_2}} \right). \tag{3.24}$$

However, lengths are increased when penalizing the inner product, *i.e.* $\left\|j'_{L_2}\right\|^2_{L_2} \leq \left\|j'_{L_2}\right\|^2_M$, such that slope in the conventional gradient direction decreases and it is therefore not the *steepest* direction anymore:

$$\{D_m J_d(m)\} \left(\frac{j'_{L_2}}{\left\|j'_{L_2}\right\|_M} \right) = \frac{\left\langle j'_{L_2}, j'_{L_2} \right\rangle_{L_2}}{\left\|j'_{L_2}\right\|_M} \tag{3.25}$$

$$= \frac{\left\langle j'_M, j'_{L_2} \right\rangle_M}{\left\|j'_{L_2}\right\|_M} \tag{3.26}$$

$$\leq \frac{\|j'_M\|_M \left\|j'_{L_2}\right\|_M}{\left\|j'_{L_2}\right\|_M} = \|j'_M\|_M = \{D_m J_d(m)\} \left(\frac{j'_M}{\|j'_M\|_M} \right). \tag{3.27}$$

Basically, the idea is to lengthen directions with undesirable features such that the slope in these directions decreases and the optimization prefers not to explore such directions. Importantly, all the model parameters are still reachable: none of them are forbidden but some are placed *a priori* further away. Increasing the strength of the penalty term reduces the misfit decrease in the preconditioned gradient direction, because the ratio $\|j'_M\|_{L_2} / \|j'_M\|_M$ appearing in (3.24) approaches zero, and it is therefore expected to slow down the convergence. A schematic illustration of an inner product modification for a two dimensional model space is given in Figure 3.2. This example illustrates two concepts: how changing the inner product modifies the descent direction and why it is useful to design the inner product based on the Hessian operator. Another schematic example is then given in Figure 3.3 to illustrate how preconditioning can modify the minima that is reached by a local optimization algorithm.

During an inversion process, the inner product is mostly used to measure model parameters update such as the gradient for example. Because of geometric spreading, shadowing or attenuation, the data gradient in full waveform inversion has a tendency to exhibit a spatial imbalance. The information to restore the balance is contained in the Hessian operator and it is therefore helpful to include this information through a spatial weight $w(\boldsymbol{x})$ in the inner product, taken as some approximation of the Hessian operator, typically the diagonal of the Gauss-Newton operator $h_{d,\mathrm{GN}}(\boldsymbol{x})$ (*cf.* Appendix C), *i.e.*

$$\langle \delta m_2, \delta m_1 \rangle_M = \int_\Omega w \, \delta m_2 \, \delta m_1 \, d\Omega + \epsilon \int_\Omega h(\delta m_2, \delta m_1) \, d\Omega. \tag{3.28}$$

This spatial weight is small in poorly illuminated regions: large updates in these regions are

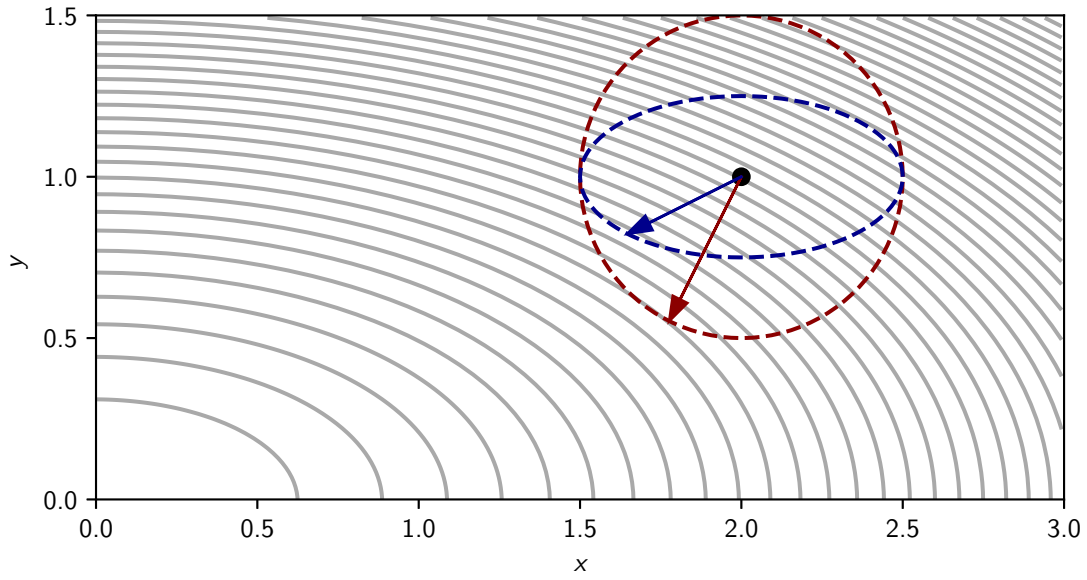


Figure 3.2: Illustration of an inner product modification for a two dimensional model space $m := (x, y)$. The performance functional is chosen as $J(x, y) = x^2 + 4y^2$ (▬, isolines). Two inner products are compared: the Euclidean norm $\langle m_a, m_b \rangle := x_a x_b + y_a y_b$ (▬) and a weighted variant $\langle m_a, m_b \rangle_M := x_a x_b + \beta^2 y_a y_b$ (▬). Both steepest descent directions (→) and unit circles (▬) are also depicted for some initial guess $m_0 = (x_0, y_0) = (2., 1.)$. The gradient in both metrics can be computed analytically from (3.18) as $j'(m_0) = (2x_0, 8y_0)$ and $j'_M(m_0) = (2x_0, 8y_0/\beta^2)$. Graphically, the steepest direction is obtained from the intersection of the unit circle with the isoline of smallest value. The second inner product penalizes updates in the y direction, because $\|m\|^2_M = \|m\|^2 + (\beta^2 - 1)y^2$ with $\beta > 1$. Hence the unit circle is flattened in the y direction and updates along the x axis are favored. Moreover, when the metric is chosen to match the Hessian, *i.e.* $\beta = 2 \Leftrightarrow P = H$, then the update direction is ideal because it points towards the minimum. Graphically, matching the metric to the Hessian means that the unit circle has the same aspect ratio than the contour isolines.

thus closer, because they have a smaller norm. The weight is however not applied on the penalization term and the smoothing term thus has a stronger influence in poorly illuminated zones, where the weight is smaller than some constant threshold ϵ.

Isotropic smoothing inner product

A first natural choice for the inner product is to take its penalization from the second order derivative of the Tikhonov additive penalization (3.3)

$$\langle \delta m_2, \delta m_1 \rangle_M = \int_\Omega w\, \delta m_2 \delta m_1 \, d\Omega + \alpha \int_\Omega \mathbf{grad}\,(\delta m_2) \cdot \mathbf{grad}\,(\delta m_1)\; d\Omega, \tag{3.29}$$

which leads to the preconditioner

$$P = w + \alpha\, \mathrm{div}\,(\mathbf{grad}\,())\,. \tag{3.30}$$

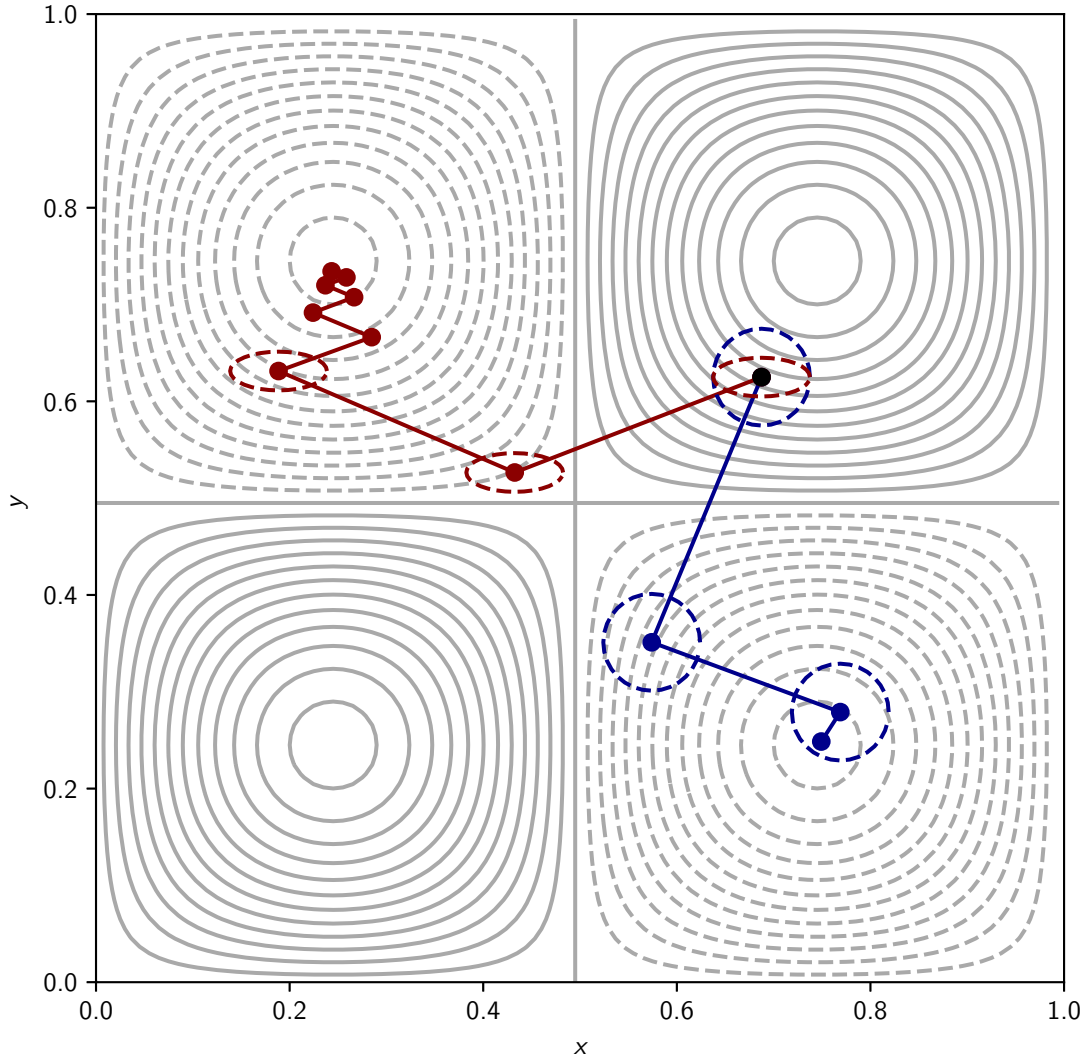


Figure 3.3: Illustration of an inner product modification for a two dimensional model space $m := (x, y)$. The performance functional is chosen as $J(x, y) = \sin 2\pi x \sin 2\pi y$ (▬, positive isolines)(▪▪, negative isolines). Two inner products are compared: the Euclidean norm $\langle m_a, m_b \rangle := x_a x_b + y_a y_b$ (▬) and a weighted variant $\langle m_a, m_b \rangle_M := x_a x_b + \beta^2 y_a y_b$ with $\beta = 3$ (▬). The line search steepest descent paths from some initial guess $m_0 = (x_0, y_0) = (11/16, 10/16)$ are depicted (▬,●). For the first three iterations, unit circles in both metrics are also depicted (▪▪). The weighted inner product favors updates along the x-axis and hence another minima is reached. Around this minima, the metric and the Hessian strongly mismatch and the convergence is thus slow, because the descent directions are alternating.

The effect of this preconditioner is thus to smooth the model update with the same strength in every direction. Here, the smoothing strength is controlled by the regularization parameter α, which is the equivalent of the regularization parameter λ for additive penalization methods. The new regularization parameter can however be linked easily to a smoothing length l_c through $\alpha = \epsilon \, (l_c / 2\pi)^2$, because it is the characteristic length of the Laplacian preconditioner

when the weight is around the threshold value $w \approx \epsilon$. Note that the threshold ϵ is now included in the definition of α: this is a slight change w.r.t. Chapter 1 and 2. Having such a physical and intuitive interpretation for this parameter is of great help in practice, because it is then possible to set reasonable bounds for its value. The lower bound is in the order of the smallest propagated wavelength while the upper bound is around the size of the imaged region. Whatever the smoothing length is, this Laplacian operator is however always isotropic. When the model parameters exhibit sharp edges, updates are still smoothed across that interface. This effect is undesirable and motivates the introduction of an *adaptative* and *anisotropic* inner product that aligns the smoothing direction along the edges. Such an inner product is introduced below.

Edge-enhancing inner product

The edge-enancing inner product is based on the edge-enhancing operator, a particular case of a structure preserving operator. These latter operators are often based on the *structure tensor* [99, 114, 193, 194, 201], which is defined as the outer product of the model spatial gradient with itself, *i.e.*

$$\boldsymbol{S}_0 := \mathbf{grad}\,(m) \otimes \mathbf{grad}\,(m)\,. \tag{3.31}$$

Its normalized version is called the parallel diffusion tensor $\boldsymbol{D}_{\|}$ because it actually projects any direction onto the local gradient direction, hence its name and notation

$$\boldsymbol{D}_{\|} := \frac{\mathbf{grad}\,(m) \otimes \mathbf{grad}\,(m)}{|\mathbf{grad}\,(m)|^2}. \tag{3.32}$$

Using this tensor, it is easy to defined an inner product with a total diffusion tensor $\boldsymbol{D}$ composed of two smoothing terms: one parallel to the gradient and one perpendicular

$$\langle \delta m_2, \delta m_1 \rangle_M = \int_\Omega w\, \delta m_2 \delta m_1 \, d\Omega + \alpha \int_\Omega \mathbf{grad}\,(\delta m_2) \cdot \boldsymbol{D} \cdot \mathbf{grad}\,(\delta m_1)\; d\Omega, \tag{3.33}$$

with

$$\boldsymbol{D} = g_{\|}(|\mathbf{grad}\,(m)|^2)\;\; \boldsymbol{D}_{\|} + g_{\perp}(|\mathbf{grad}\,(m)|^2)\;\; \boldsymbol{D}_{\perp} \tag{3.34}$$

where $\boldsymbol{D}_{\perp}$ and $\boldsymbol{D}_{\|}$ are again the perpendicular and parallel structure tensors while $g_{\perp}$ and $g_{\|}$ are the perpendicular and parallel diffusion amplitudes, that still needs to be chosen. The associated preconditionner is again a Laplacian operator but it is now spatially inhomogenous, model parameter dependent and anisotropic, *i.e.*

$$P = w + \alpha\, \mathrm{div}\,(\boldsymbol{D}(\mathbf{grad}\,(m)) \cdot \mathbf{grad}\,())\,. \tag{3.35}$$

The anisotropy comes from the difference between the perpendicular diffusion amplitude $g_{\perp}$ and the parallel diffusion amplitude $g_{\|}$ while the adaptativity is obtained because both the

diffusion amplitudes and both the diffusion tensors depend the current model parameter gradient. Taking both diffusion amplitudes to be equal, *i.e.* $g_{\|} = g_{\perp}$, gives an adaptative isotropic smoothing because $\boldsymbol{D}_{\|} + \boldsymbol{D}_{\perp} = \boldsymbol{I}$ and it reduces to the constant isotropic inner product (3.29) when both diffusion amplitudes are set to one. The diffusion term of this general inner product actually encapsulates the second order derivative of the total variation misfit functional (3.15). It suffices to take $g_{\perp}(r^2) = 1/r$ and $g_{\|}(r^2) = 0$, but this is not the best option. Firstly, the related smoothing is *never* isotropic, even in very flat regions of the model parameter. In addition, in these flat regions, the smoothing length tends to infinity. Secondly, the penalization term totally disappears on edges, where the gradient is large. A more satisfying option [193, 194] is to choose the diffusion amplitudes to satisfy the two asymptotic behaviors $g_{\|}(r^2 \ll \beta^2) = 1$ and $g_{\|}(r^2 \gg \beta^2) = 0$; the transition between both regimes being set by the *edge threshold* parameter β. With such a choice, the smoothing is parallel to any sharp interface ($|\mathbf{grad}\,(m)| \gg \beta$) and isotropic in flat regions ($|\mathbf{grad}\,(m)| \ll \beta$). In this chapter, a very simple choice is selected

$$g_{\perp}(r^2) = 1 \quad \text{and} \quad g_{\|}(r^2) = \frac{1}{1 + r^2/\beta^2} \tag{3.36}$$

such that the total diffusion tensor reduces to

$$\begin{aligned} \boldsymbol{D} &= g_{\|}(|\mathbf{grad}\,(m)|^2)\ \ \boldsymbol{D}_{\|} + \boldsymbol{D}_{\perp} && (3.37)\\ &= \boldsymbol{I} - (1 - g_{\|}(|\mathbf{grad}\,(m)|^2))\ \ \boldsymbol{D}_{\|} && (3.38)\\ &= \boldsymbol{I} - \frac{\mathbf{grad}\,(m) \otimes \mathbf{grad}\,(m)}{|\mathbf{grad}\,(m)|^2 + \beta^2}. && (3.39) \end{aligned}$$

Again, it is straightforward to see from this reduced expression that the total diffusion tensor is perpendicular to the gradient when it is large and isotropic when it is small. It is interesting to mention that the β parameter involved in the derivatives of the total variation penalization and the edge threshold parameter β introduced here above both appear mathematically at the same place, *i.e.* both are added to the model gradient, hence the shared symbol β. Nevertheless, their values are chosen to satisfy very different purposes: the former is very close to zero and aims to prevent numerical instabilities while the latter is larger and is used to discriminate edges from slowly varying regions. The last step of this structure preserving inner product is to use a smoothed structure tensor $\boldsymbol{S}_{\rho}$ instead of the regular structure tensor $\boldsymbol{S}_0$. This smoothing, at the *structure scale* ρ, makes the structure tensor robust against reconstruction artifacts, and therefore allows a more reliable estimation of orientations. Moreover, it spreads the orientation information into the regions between edges and thus it allows to estimate the dominant orientation also at those points in the reconstruction where the gradient is close to zero [20]. At this point, one can wonder why the whole approach is based on quadratic gradient forms rather than the gradient itself. It is actually because smoothing the gradients

can lead to cancellation effects. Indeed, consider for example a thin line-like structure: on one side of the peak the gradient is positive, while on the other side the gradient is negative (*cf.* Figure 3.1c).

In summary, this structure preserving inner product depends on two *a priori* parameters: the edge threshold β and the structure scale ρ. Their values of course depend on the nature of the model parameters to be inverted and making a choice is a way to introduce prior information in the inversion. When prior information about the size of the anomalies to be imaged is not available, the structure scale can be set to approximately half the smallest wavelength, because no smaller structures are expected to be imaged, even without noise [171]. Similarly, when prior information about model parameter jumps is missing, the interface threshold β can be selected such that the identified edges in the initial distribution are slightly above the threshold. The amplitude of the gradient and the threshold are shown graphically in Figure 3.4.

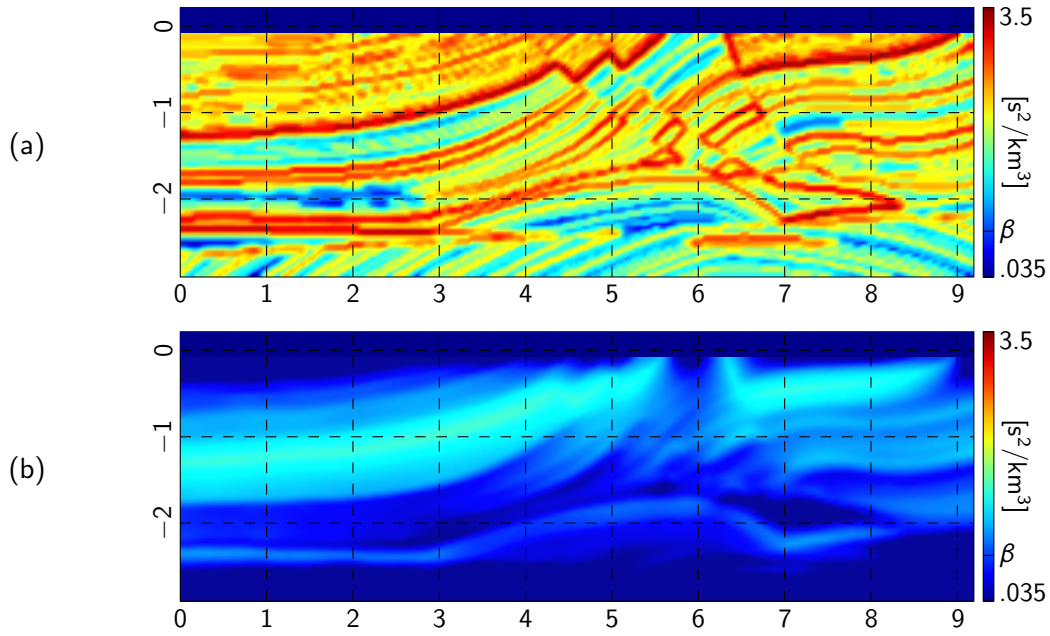


Figure 3.4: Norm of the slowness squared gradient $|\mathbf{grad}\,(s^2)|$, filtered at the structure scale $\rho = 0.125$ [km], for the ground-truth distribution (a) and for the initial distribution (b). The chosen edge threshold parameter β is given graphically. Color scale is logarithmic.

The regularization parameter α is sightly different: it governs the strength of the penalization but not the penalization itself. It is the equivalent of the regularization parameter λ for additive penalization methods.

In the case of multi-parameter inversion, the approach can be generalized trivially by computing one diffusion tensor per parameter class. However, in many natural environments, the structure of the different model parameters is shared, as jumps in model parameters coincide with jumps of the underlying medium type. The situation is the same than stating that the jumps in the RGB channels of a colour image are correlated because there is a single underlying

object [195]. When it can be considered that there is a single coherent structure for all the parameter classes, then a single structure tensor must be used. Sharing the structure tensor allows to inject information form one parameter class to another, hence it brings an additional soft constraint. To build this structure tensor, the idea is then to perform a weighted mean between the individual structure tensors

$$\boldsymbol{S}_\rho = \sum_i q_i \boldsymbol{S}_\rho(m_i) \quad \text{with} \quad \sum_i q_i = 1. \tag{3.40}$$

The weight q_i given to each tensor should then reflect the relative confidence given to the individual parameter reconstruction. For example, if some parameter is poorly constrained by the data, then it should have a small impact on the structure tensor [195]. Interestingly, this approach introduces a coupling between parameter classes but the preconditioner can still be applied to each parameter class independently at each inner iteration, because the coupling appears at the outer iteration level. This is one of the main difference with recent applications of diffusion filtering for full waveform inversion [114, 204], where the non-linear diffusion process is applied to the gradients, at the inner iterations level. Here, the somehow equivalent diffusion process (3.19) rather appears at the outer iteration level, which allows the diffusion tensor to be constructed from the current model parameters class-wise, rather than from the class-by-class gradients.

3.2.3 Link between both approaches

Similarities between both approaches have been hinted during their introduction: they are made explicit in this subsection. The link between both regularization strategies comes from the similarity, by construction, of the Hessian operator from the additive penalization method and the preconditioner from the inner product method

$$H = H_d + \lambda \mathrm{div}\left(\boldsymbol{D} \cdot \mathbf{grad}\left(\right)\right) \quad \text{and} \quad P = h_{d,\mathrm{GN}} + \alpha\, \mathrm{div}\left(\boldsymbol{D} \cdot \mathbf{grad}\left(\right)\right). \tag{3.41}$$

Obviously, the preconditioner operator is an approximation of the Hessian operator. The reason for that is simple: thanks to this choice, the preconditioned gradient is getting closer to the penalized misfit Newton direction, *i.e.*

$$j'_M = P^{-1} j'_{d,L_2} \quad \sim \quad p = H^{-1}(j'_{d,L_2} + \lambda j'_m), \tag{3.42}$$

while breaking free from some of its drawbacks. Indeed, the regularization part of the Hessian H is dictated by the misfit penalization J_m through its second order derivative $D^2_{mm} J_m$, with no degree of freedom. It is therefore impossible to decouple the preconditioning effect that appears in the Hessian H from the penalization effect that appears in the misfit. At the opposite, inner product modification does not involve any other misfit than the data misfit

and hence the regularization parameter α can be chosen freely. For example, to produce a strong smoothing, the regularization parameter λ must be large but then the data part of the gradient becomes negligible. Inner product modification acts exclusively as a preconditioner on the update directions, without changing the misfit. Preconditioning can then be adapted freely to produce the best update direction, without misfit related constraints. In particular, it can have a strong preconditioning effect while still being driven only by the data misfit.

3.3 Reference case studies

For the numerical case studies, the same general setting as in Chapter 1 and 2 is considered. The only difference lies in the synthetic data generation. In this chapter, measured data are generated in the time domain, using a Newmark time stepping scheme ($\gamma = 0.5$, $\beta = 0.25$) [128] and a Ricker excitation of center frequency f_c

$$f(t) = (1 - 2\tau^2)\exp(-\tau^2) \quad \text{with} \quad \tau = \pi f_c t. \tag{3.43}$$

Then a band limited Gaussian noise $n(t)$ is added to these data before the extraction of the frequency component to be inverted; hence avoiding any inverse crime. Noise levels will be specified in terms signal-to-noise ratio

$$\text{SNR} := \frac{\int_0^\infty |u(t)|^2 \, dt}{\int_0^\infty |n(t)|^2 \, dt}. \tag{3.44}$$

In this synthetic setting, several criterion can be used to select the best regularization parameters or the best stopping iteration. The ideal criterion should be based on some error w.r.t. the ground-truth $\tilde{m}$, *e.g.* the mean relative error

$$\frac{1}{A_\Omega} \int_\Omega \frac{|m - \tilde{m}|}{|\tilde{m}|} \, d\Omega, \tag{3.45}$$

with $m = s^2$ or $m = (\epsilon_r, \sigma_r)$ depending on the case study. Of course, outside any synthetic setting, this error can not be computed as the ground-truth is unknown. A more realistic approach is to use the so-called *discrepancy principle* [180], which can be used when an estimation of the noise level is available. This simple criterion proposes to select the regularization parameter λ or the stopping iteration N such that the data misfit drops to the noise level

$$J_d(m) = \frac{1}{2} \|d(m) - d_{0,n}\|^2 \approx \frac{1}{2} \|d_0 - d_{0,n}\|^2 = J_d(\tilde{m}). \tag{3.46}$$

The motivation behind this criterion is straightforward: a smaller misfit would necessarily mean that the data are fitting some part of the noise. For real applications however, the noise

level is unfortunately not available and other techniques should be used instead. The most widely used is probably the L-curve method [75], but in the context of full waveform inversion, pragmatic rules are also often used. The simplest strategy is to limit the computational cost of the method, typically by stopping the inversion algorithm after a fixed number of iterations. Such a practical strategy can easily be implemented with a preconditioning strategy, but it is harder to implement with an additive misfit strategy, because one value of the regularization parameter λ requires a complete, fully converged, inversion.

For the remainder of this chapter, the *l*-BFGS algorithm combined with a line search procedure is used for the minimization. The same weighted and thresholded inner product is used for both regularization strategies, but of course a second term is added for the inner product preconditioning method.

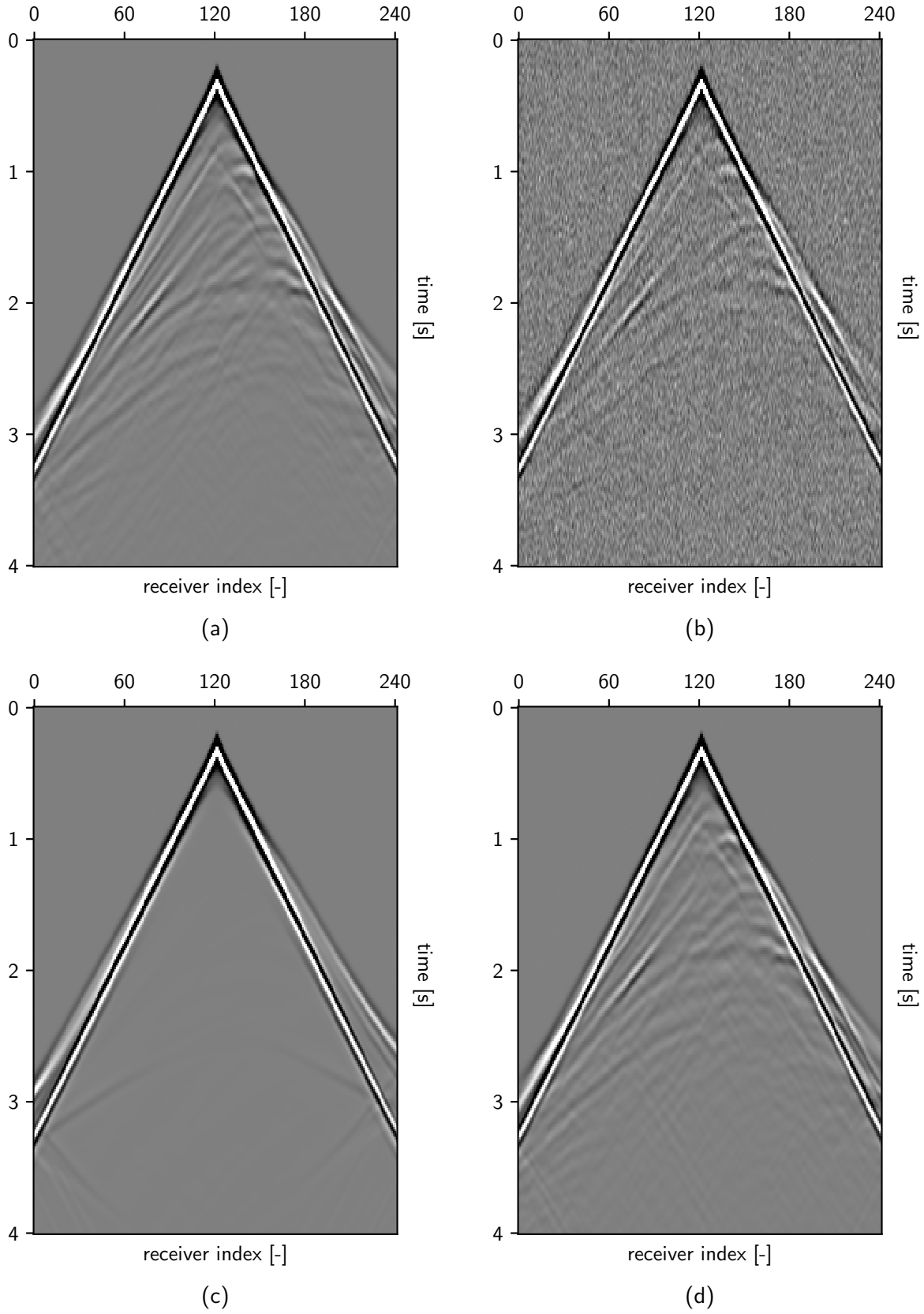


Figure 3.5: Time domain data for a single source, computed using a Newmark time-stepping scheme ($\gamma = 0.5$, $\beta = 0.25$, $\Delta t = 4$ [ms], $T = 8$ [s]) for different slowness squared distributions. Ground-truth (a). Ground-truth with band limited Gaussian noise (SNR $= 4.$ [-]) (b). Initial distribution (*cf.* Figure 2.4b) (c). Final distribution (*cf.* Figure 3.6c) (d).

3.3.1 Case study 1: Marmousi model

As previously mentioned, data are now generated in the time domain, with a Ricker excitation ($f_c = 6$ [Hz]). A long time window ($T = 8$ [s]) is simulated with a small time steps ($\Delta t = 4$ [ms]). Hierarchical finite elements of order 2 are used for the spatial discretization of the pressure fields. A band limited ($B = 15$ [Hz]) Gaussian noise (SNR $= 4$ [-]) is then added to these perfect data before computing their Fourier transform at the three frequencies used for the inversion ($f = 4, 6, 8$ [Hz]). The ground-truth data and the noisy data in the time domain for a single shot are given in Figure 3.5a and 3.5b. In addition, time data corresponding to the initial and a typical final model parameter are also given in 3.5c and 3.5d.

Additive penalization Inversion results have been computed for both the Tikhonov penalization and the total variation penalization, and for several values of the regularization parameter λ. In these cases, the iterative process is stopped after a maximum number of 60 iterations, or as soon as the misfit does not decrease significantly anymore. Final inversion results for the best values of the regularization parameter λ are given in Figure 3.6. The final *relative* data misfit, *i.e.* the final data misfit divided by the data misfit associated to the ground-truth model, *i.e.* $J(s^2)/J(\tilde{s}^2)$, is given in Figure 3.7a for several values of the regularization parameter. The relative misfit is used instead of the absolute misfit because it is easier to interpret. In the sense of the discrepancy principle, the reconstruction under-fits (resp. over-fits) the data when this relative misfit is greater (resp. smaller) than one. The mean relative slowness squared error w.r.t. the ground-truth is given Figure 3.7b, also for several values of the regularization parameter. As can be seen from Figure 3.7b, selecting the regularization parameter according to the discrepancy principle is not optimal. The discrepancy principle actually over-smooths the inversion results as it stops too early. Fitting more noise but also more data actually gives better reconstruction [75]. Both penalization strategies give smaller errors than the unpenalized methods (*cf.* Table 3.1) even though the total variation penalization is slightly more efficient. Both actually succeed to suppress some of the noise artifacts. The Tikhonov penalization yields a smoother image while the total variation penalization gives a more staircase like reconstruction [27]. Both regularized inversions still feature some artifacts, even if they are smaller than for the unregularized inversion. The optimal value for the regularization parameter is actually very small and the penalization term is inactive during the first few iterations of the optimization algorithm. Comparing the early reconstructions of all three algorithms actually shows that all are very similar (*cf.* Figure 3.8), because differences only appear in the end, when the additive penalization becomes active. Such a behavior can be problematic if the iterative algorithm is stopped before convergence, because the removal of noise artifacts only appears in the end, when reaching a balance between the data misfit term and the penalization term. For example, the constant velocity structure at $(x, y) = (7, -2)$ [km] features oscillations during early iterations (Figure 3.8b) but is homo-

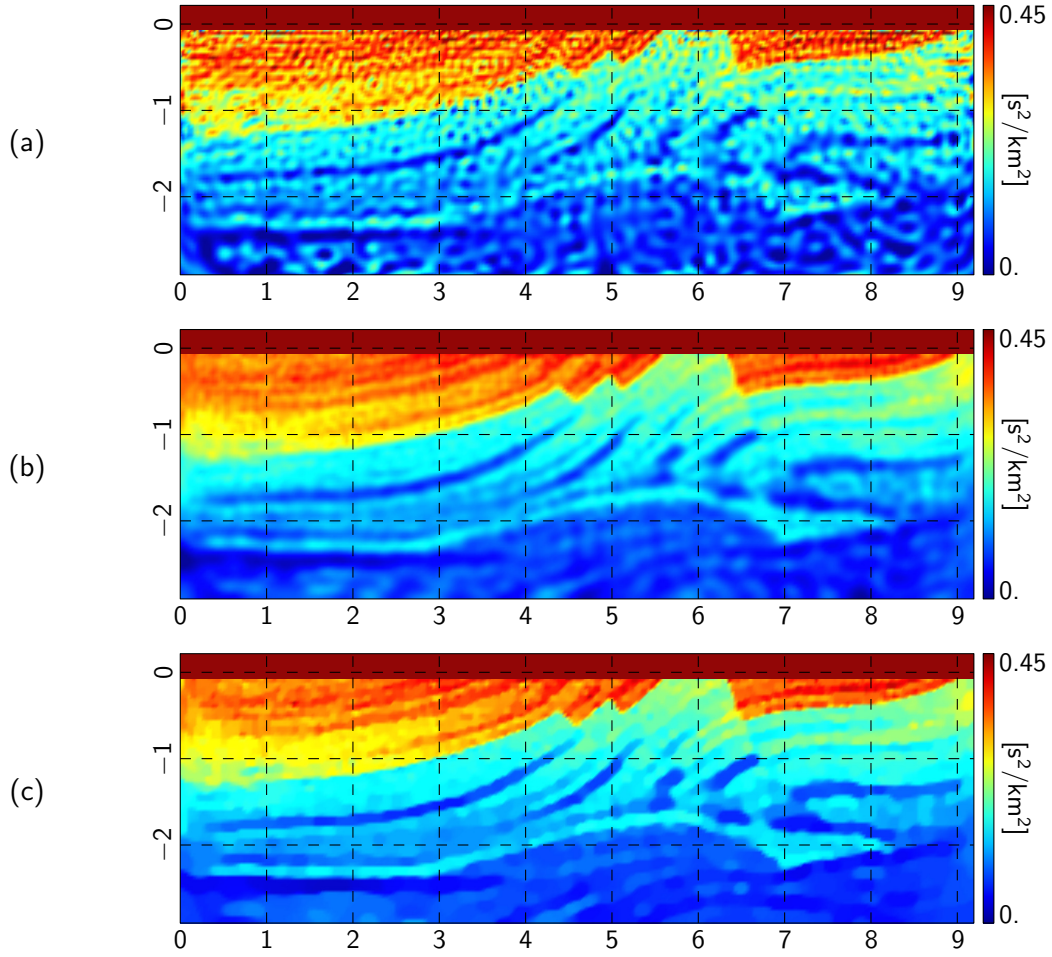


Figure 3.6: Final inversion results using additive penalization regularization. No penalization (a). Tikhonov penalization (b). Total variation penalization (c). Regularization parameters λ are chosen as the best in the sense of the slowness squared error.

geneous in the final model (Figure 3.6c). Interestingly, for the Tikhonov penalisation, there is a formal equivalence between the regularization parameter λ and the regularization parameter α from the isotropic inner product. Then because α is linked to a characteristic length, it is possible to relate λ to a characteristic length

$$\lambda \leftrightarrow \alpha = \epsilon(l_c/2\pi)^2 \quad \Rightarrow \quad l_c = 2\pi\sqrt{\lambda/\epsilon}. \tag{3.47}$$

Using the best value of the regularization parameter λ, this length can be computed as $l_c \approx 30$ [m], which is much smaller than what is typically used for the preconditioners, as is showed in the following section.

Inner product preconditioning Using preconditioning as the only regularization strategy, the iteration number becomes the formal regularization parameter [86]. All the other inner product parameters should therefore be set *a priori*. The relative importance of the smoothing inner product term is set through the regularization parameter α, which is linked to a

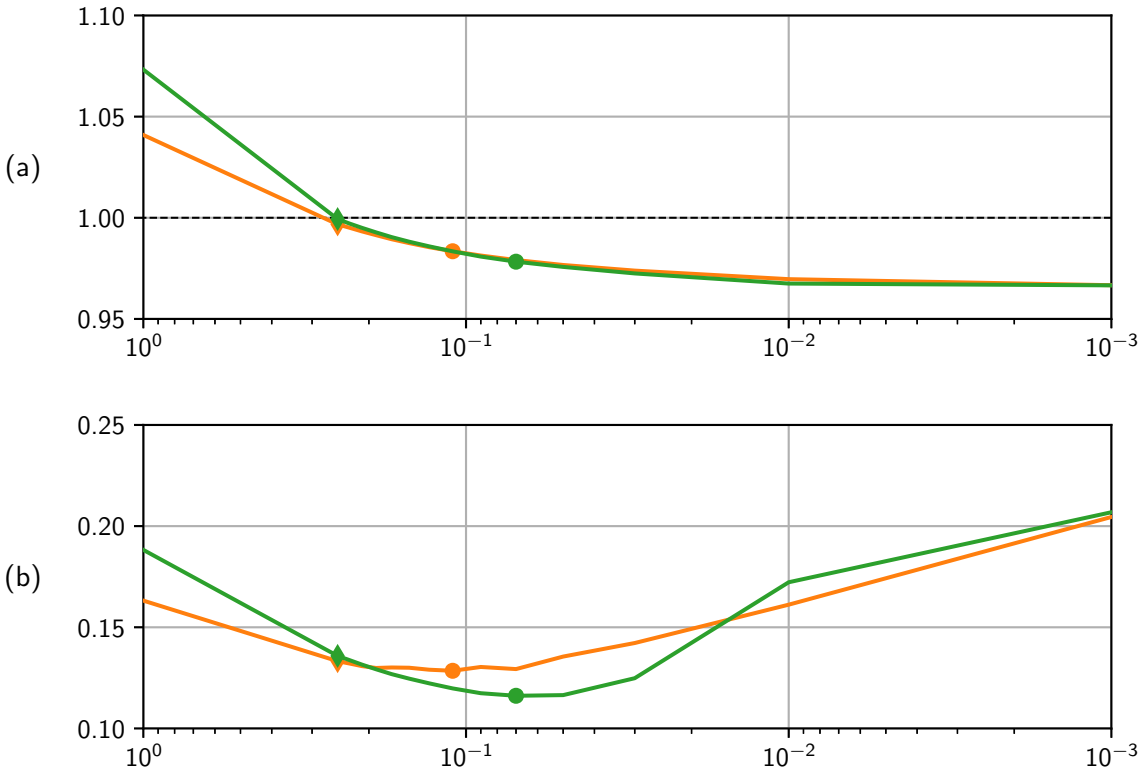


Figure 3.7: Final relative data misfit (a) and mean relative error w.r.t. the ground-truth (b) for several values of the regularization parameter λ and for both Tikhonov penalization (▬) or total variation penalization (▬). Markers indicate the best regularization parameters in the sense of the discrepancy principle (♦) and in the sense of the minimal error (●).

	-	TIK	TV	ISO-IP	ANISO-IP
Minimal error	0.148	0.129	0.116	0.14	0.126
Discrepancy	0.155	0.133	0.136	0.148	0.134

Table 3.1: Mean relative error at the best iteration and/or for the best regularization parameters in the sense of the discrepancy principle and in the sense of the minimal error for different regularization strategies: Tikhonov (TIK) and total variation (TV) additive penalizations; isotropic (ISO-IP) and anisotropic (ANISO-IP) inner product preconditioning.

characteristic smoothing length l_c. Here $l_c = 0.5$ [km] and $l_c = 2$ [km] are chosen for the isotropic and anisotropic case respectively. This smoothing length can be taken bigger for the adaptative variant because it is able to tilt the smoothing direction and hence sharp features are still reachable reasonably fast, even with such a large smoothing length. These two choices are reasonable as far as the scales involved in the Marmousi case study are concerned: the minimum propagated wavelength is around 0.250 [km] while the acquisition size is around 9 [km]. For now, both these lengths are fixed but their influence is studied later. As far as the edge-enhancing inner product is concerned, the structure scale is set to $\rho = 0.125$ [km], which

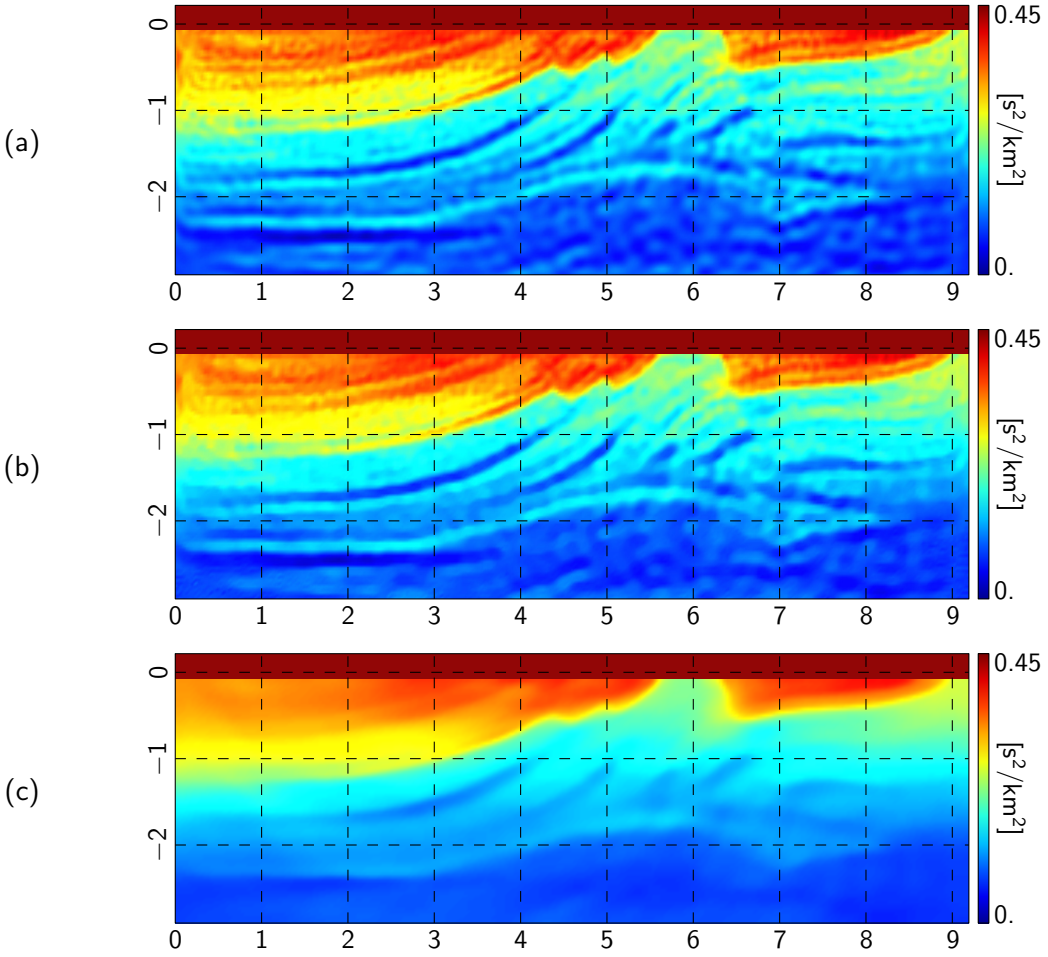


Figure 3.8: Early inversion results (7th iteration). No regularization (a). Total variation additive penalization (b). Edge-enhancing inner product preconditioning (c). Regularization parameters λ and α are chosen as the best in the sense of the relative error.

is approximately half the smallest wavelength, while the interface threshold β is shown graphically in Figure 3.4. It is taken such that the initial distribution has very few identified edges, *i.e.* $g_{\|}(s^2_{\text{init}}) \approx 1$, but such that all the edges of the ground-truth distribution are detected, *i.e.* $g_{\|}(\tilde{s}^2) \approx 0$. Reconstructions at the optimal iteration in the sense of the mean relative error are given in Figure 3.9, while the relative misfit and this error are given in Figure 3.10a and 3.10b for all the successive iterations. As expected, the edge-enhancing inner product yields particularly accurate inversion results, as can be seen from Figure 3.9 and from the corresponding relative error (*cf.* Table 3.1). In particular, nearly no noise artifacts are present in the image. The linear structures in the upper left corner are much better reconstructed with the anisotropic inner product. Its convergence speed in the data space is slightly slower, because the penalization is stronger. In the model space however, the error convergence is faster from the 15th iteration and this is actually what really matters. With a hypothetical perfect preconditioner, the model error should be zero when the discrepancy principle is satisfied. Starting from the standard inner product, the isotropic inner product is one step towards that perspective and the anisotropic variant is one step further, even though it is still

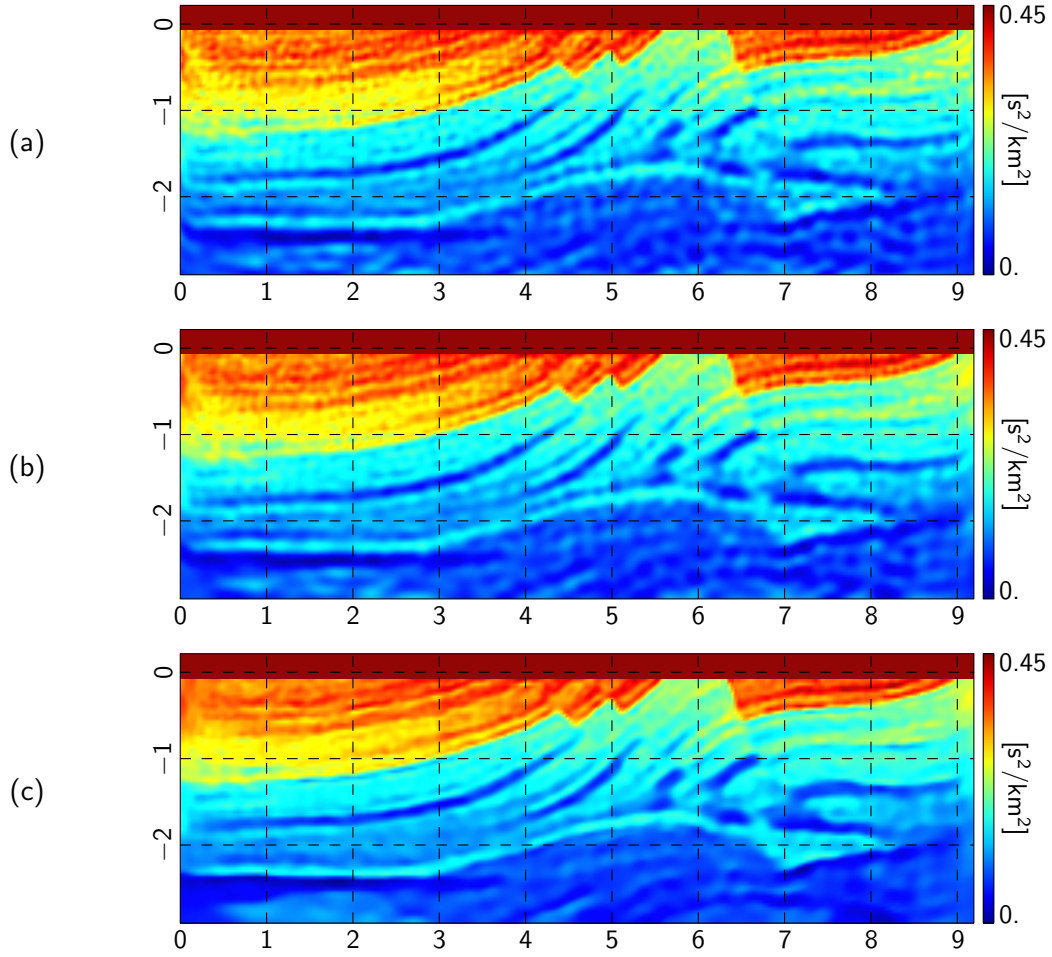


Figure 3.9: Final inversion results using inner product preconditioning regularization. No smoothing term (a). Isotropic smoothing term (b). Anisotropic smoothing term (c). Stopping iterations are chosen as the best in the sense of the mean relative error.

far from being the perfect choice. Preconditioning has an influence from the beginning of the optimization algorithm, as can be seen from the early slowness square reconstruction, which is actually much smoother than its additive penalization counterpart (*cf.* Figure 3.8). However, preconditioning fails to prevent the apparition of noise artifacts at the end of the optimization, as witnessed by the increasing error tail in Figure 3.10b and it is therefore critical to stop the iterative process soon enough. Unfortunately, identifying this stopping point is not easy: the discrepancy criterion is again not close from the optimal iteration and even if it were, the noise level is still unknown in most applications. The bottleneck with preconditioning techniques is to find this optimal stopping iteration. After that point, the error starts to increase because of inevitable noise fitting. In a context where the computational cost is limited, the stopping iteration is however often chosen much smaller than the optimal iteration, hence the interest for the method.

The only left question to tackle is now how to choose the regularization parameter α or equivalently the smoothing l_c. To understand the influence of this parameter, the inversion algorithms have been launched for several characteristic length l_c. The corresponding error

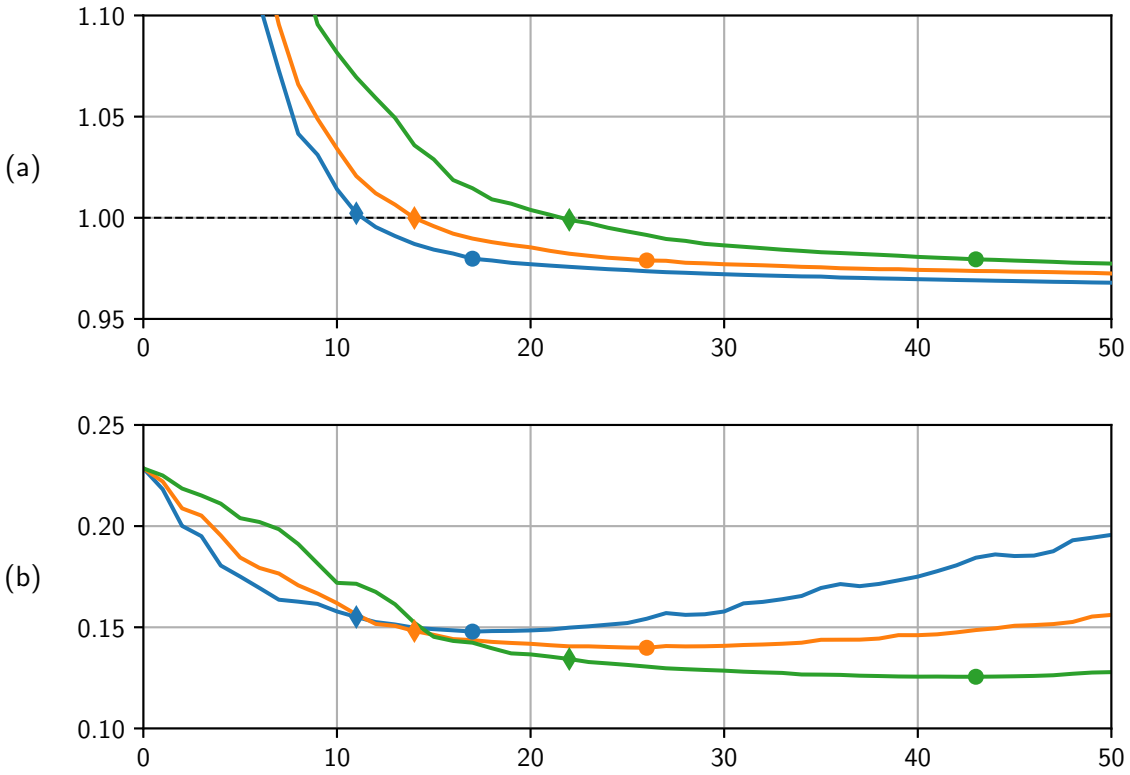


Figure 3.10: Relative data misfit (a) and mean relative error (b) for all the successive iterations, for an isotropic (▬), an anisotropic (▬) and a weighted only (▬) inner product. Markers indicate the best iterations in the sense of the discrepancy principle (♦) and in the sense of the minimal error (●).

histories are given in Figure 3.11a-3.12a while the relative misfit histories are given in Figure 3.11b-3.12b. Both figures summarized in Table 3.2-3.3. The relative misfit decrease is also plotted in Figure 3.11c-3.12c against the (dimensionless) inner product penalty

$$\frac{1}{A_\Omega\,\beta^2}\frac{\|m\|_M^2-\|m\|_{L_2}^2}{\alpha}=\frac{1}{A_\Omega\,\beta^2}\int_\Omega \mathbf{grad}\,(m)\cdot\boldsymbol{D}\cdot\mathbf{grad}\,(m)\;d\Omega. \tag{3.48}$$

Such a plot of the data misfit versus a model-based penalty is often called a L-curve, because of its shape [75].

The first observation is that the inversion results actually depend on this regularization parameter α. However, taking the smoothing length between $l_c = 1$ [km] and $l_c = 8$ [km] gives reasonable inversion accuracy for the edge-enhancing inner product, similar than the penalization counterpart. The isotropic inner product seems to be more sensitive to the choice of the smoothing length. It fails to converge in the model space when it is too large. This is because the smoothing direction does not change, hence the inversion troubles to reconstruct the sharp variation in depth. Whatever the regularization length, the anisotropic variant converges faster in the model space than the isotropic variant, as the discrepancy limit is systematically reached in less iterations and gives a lower error. When two smoothing lengths

give the same reconstruction quality, it is always better to choose the smallest, because a larger smoothing length always implies a slower convergence in the data space and at some point (for $l_c > 2$ [km]), also a slower convergence in the model space. Practically, the regularization parameter α is hard to set *a priori*, but the fact that it is related to a smoothing length restricts the interval of interesting values to explore. When it is too small, typically the order of the wavelength then the smoothing has nearly no effect and when it is too large, typically bigger than the size of the domain then the smoothing is too strong, because any direction is nearly transformed into a constant distribution. An efficient qualitative strategy to choose this parameter is the L-curve (*cf.* Figure 3.12c-3.12c). Starting from the largest characteristic length l_c and going to the smallest, the L-curves are moving to the left, which means that the same data misfit can be obtained for a smaller model penalization. For a given data misfit level, *e.g.* the discrepancy limit, it is thus easy to choose between two regularization parameters: the regularization parameter which gives the smoother model, in the sense of the penalization (3.48), should be chosen. Nevertheless, it should be kept in mind that increasing the strength of the inner product penalty decreases the convergence speed. Hence the smoothing length should not be increased anymore as soon as the L-curve start superposing. This L-curve shifting effect is particularly visible for the anisotropic penalisation, because the penalization is more suited to the Marmousi model structure. Using the argument developed here above, the smoothing $l_c = 1$. [km] and $l_c = 2$. [km] would have been selected. In summary, misfit penalization methods and preconditioned methods have the same reconstruction potential as far as this first case study is concerned. The strategy for noise removal is however very different. Preconditioning methods remove the noise at the beginning of the inversion while misfit penalization methods suppress the noise in the end of the inversion. It could therefore be tempting to combine both methods, but it would be useless in this case: if the algorithm is stopped before convergence than the misfit penalization is inactive and if the algorithm is carried to full convergence than there is no influence of the preconditioned path, provided the same minimum is reached. This latter condition is however not always fulfilled, as is shown in the next case study.

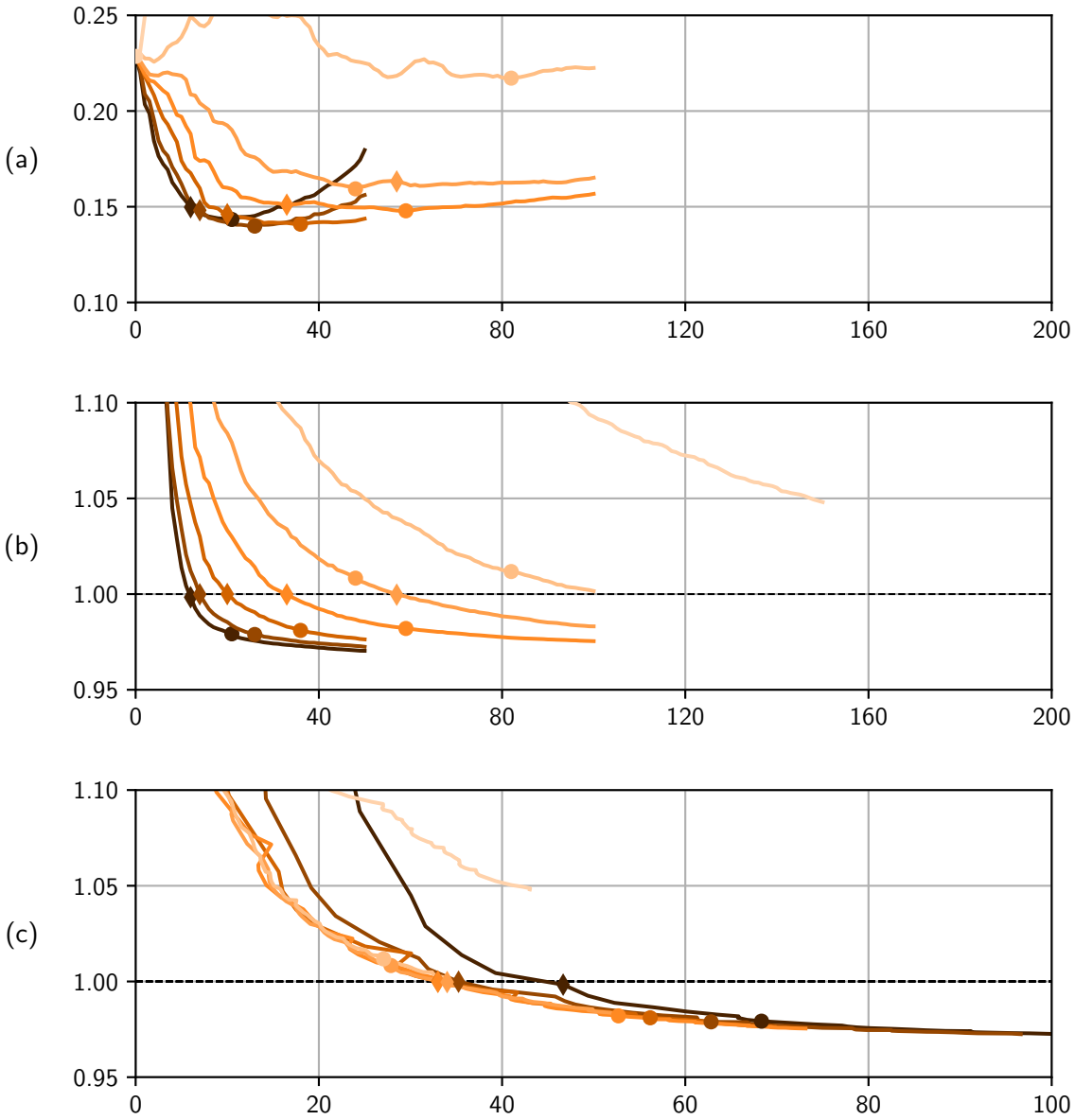


Figure 3.11: Mean relative error against the iteration number (a). Relative misfit against the iteration number (a) and against the inner product penalty (c). The anisotropic inner product is used. Markers indicate the best iterations in the sense of the discrepancy principle (♦) and in the sense of the minimal error (•). $l_c = 0.25$(▬), 0.5(▬), 1(▬), 2(▬), 4(▬), 8(▬), 16(▬)[km].

l_c	0.25	0.5	1.	2.	4.	8.	16.
Minimal error	0.143	0.140	0.141	0.148	0.159	0.217	0.229
iteration #	21	26	36	59	48	82	0
Discrepancy	0.150	0.148	0.146	0.151	0.163	-	-
iteration #	12	14	20	33	57	-	-

Table 3.2: Mean relative error at the best iterations in the sense of the discrepancy principle and in the sense of the minimal error for inversions with a isotropic inner product with different smoothing lengths.

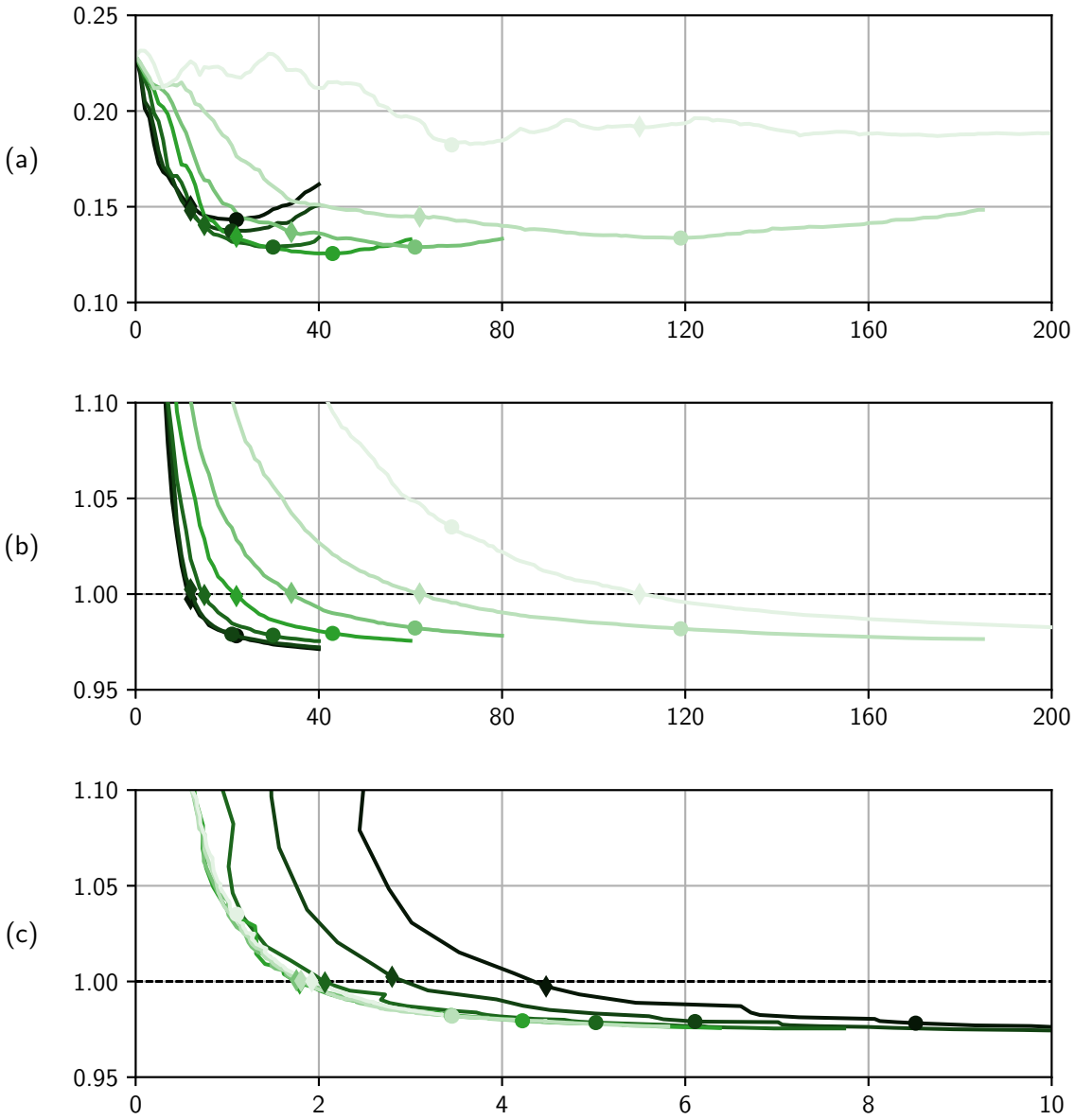


Figure 3.12: Mean relative error against the iteration number (a). Relative misfit against the iteration number (a) and against the inner product penalty (c). The anisotropic inner product is used. Markers indicate the best iterations in the sense of the discrepancy principle (♦) and in the sense of the minimal error (•). $l_c = 0.25$(▬), 0.5(▬), 1(▬), 2(▬), 4(▬), 8(▬), 16(▬)[km].

l_c	0.25	0.5	1.	2.	4.	8.	16.
Minimal error	0.143	0.137	0.129	0.126	0.129	0.134	0.182
iteration #	22	21	30	43	61	119	69
Discrepancy	0.15	0.148	0.14	0.134	0.137	0.145	0.192
iteration #	12	12	15	22	34	62	110

Table 3.3: Mean relative error at the best iterations in the sense of the discrepancy principle and in the sense of the minimal error for inversions with an anisotropic inner product with different smoothing lengths.

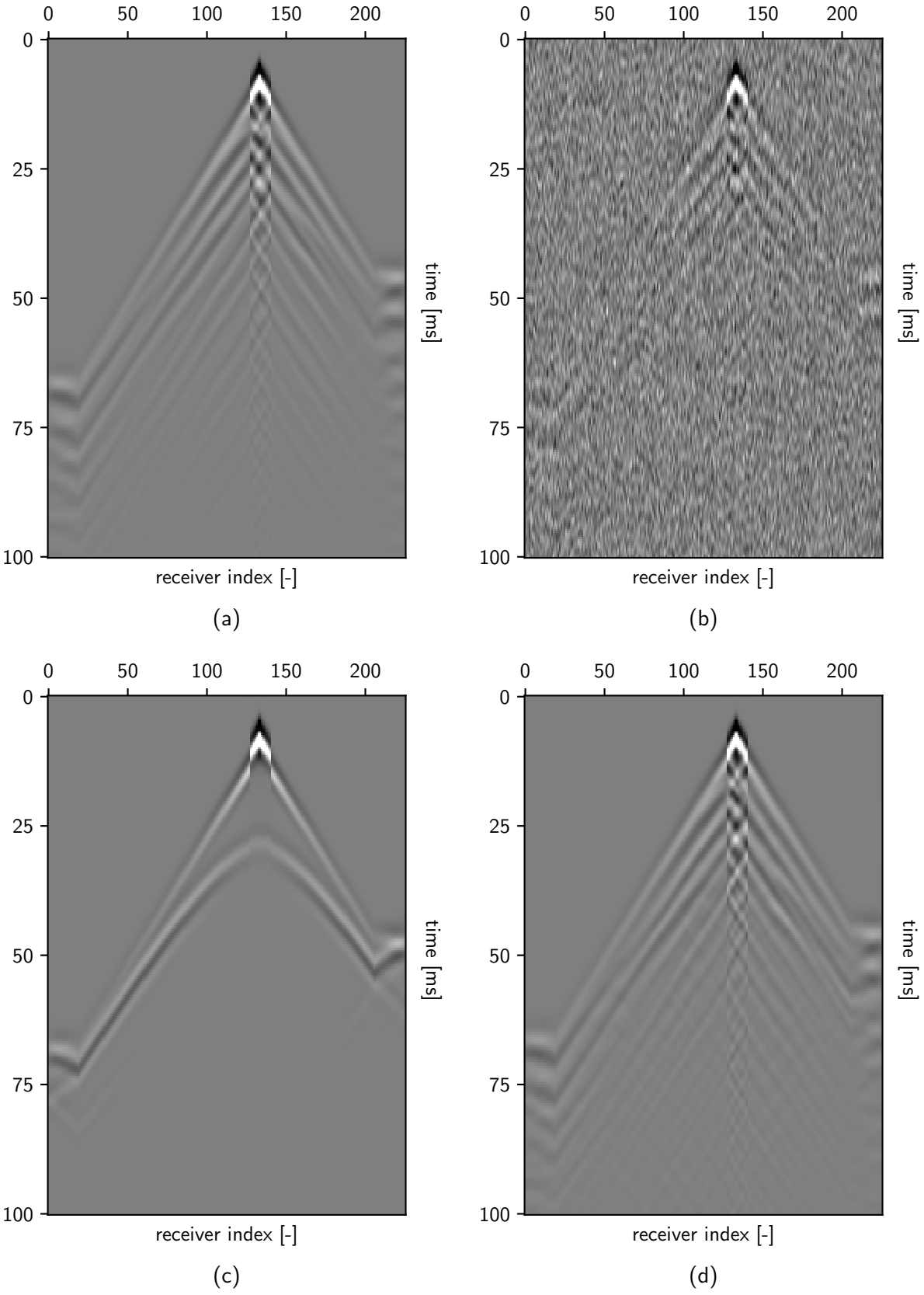


Figure 3.13: Time domain data for a single source, computed using a Newmark time-stepping scheme ($\gamma = 0.5$, $\beta = 0.25$, $\Delta t = 0.1$ [ms], $T = 0.2$ [s]) for different slowness squared distributions. Ground-truth (a). Ground-truth with band limited Gaussian noise (SNR $= 0.125$ [-]) (b). Initial distribution (*cf.* Figure 2.11a) (c). Final distribution (*cf.* Figure 3.15d) (d).

3.3.2 Case study 2: T-shaped reflectors

Similarly to the first case study, data are again generated in the time domain ($f_c = 150$ [Hz], $T = 0.2$ [s], $\Delta t = 0.1$ [ms]) with hierarchical finite element of the order 2 for the spatial discretization. A band limited ($B = 375$ [Hz]) Gaussian noise (SNR $= 0.125$ [-]) is then added to these perfect data before computing their Fourier transform at the nine frequencies used for the inversion (from $f = 100$ [Hz] to $f = 300$ [Hz] with 25 [Hz] increment). The ground-truth data and the noisy data in the time domain for a single shot are given in Figure 3.13a and 3.13b. In addition, time data corresponding to the initial and a typical final model parameter are also given in 3.13c and 3.13d. For this case study, the weight is chosen to be constant, even the part related to the algorithmic change of variable: this is a slight change w.r.t. Chapter 2. This constant is set in the same order of magnitude than the diagonal of the Gauss-Newton hessian corresponding to the initial model. Note that throughout all this case study, error are always compute on the slowness squared s^2, even though the minimization problem uses a logarithmic parametrization.

Additive penalization Inversion results have again been computed for both the Tikhonov penalization and the total variation penalization, and for several values of the regularization parameter λ. The iterative process is stopped after a maximum number of 250 iterations, or as soon as the misfit does not decrease significantly anymore. The inversion results for the best regularimzation parametre λ are given in Figure 3.15a-3.15a while the model error and the data misfit are given as solid lines in Figure 3.14a and 3.14b. As the regularization parameter λ decreases, the data misfit also initially decreases but for small values, it reaches a plateau or slightly increases and the model error explodes. For these small values, the penalization is not constraining the model parameter strongly enough and reconstructions are then similar to the unregularized reconstruction from Chapter 2, *e.g.* Figure 2.11c). For larger values, the amplitude of the velocity anomaly are strongly under-estimated (*cf.* Figure 3.15a-3.15a). To better understand what is happening, it is interesting to use preconditioning to solve the same optimization problems. The Tikhonov penalized misfit is combined with the isotropic inner product with $l_c = 5$ [m] while the total variation penalized misfit is combined with the anisotropic inner product with $l_c = 9$ [m]. The optimization path is thus different, but the global minimum remains the same because the performance functional is unchanged. The model error and the data misfit for the preconditioned inversions are given as dashed lines in Figure 3.14a and 3.14b. The behavior is now very different. In particular, the error curve is now much lower for small regularization parameters. This shows that the minimizer found by the preconditioned method is different: for the same performance functional, the minimum that is reached is different. Another way to see this is to look at the (relative) total performance functional, *i.e.* $\left(J_d(m) + J_m(m)\right) / J_d(\tilde{m})$, which is given in Figure 3.14c for both preconditioned and un-preconditioned inversions. If the same minimizer were reached,

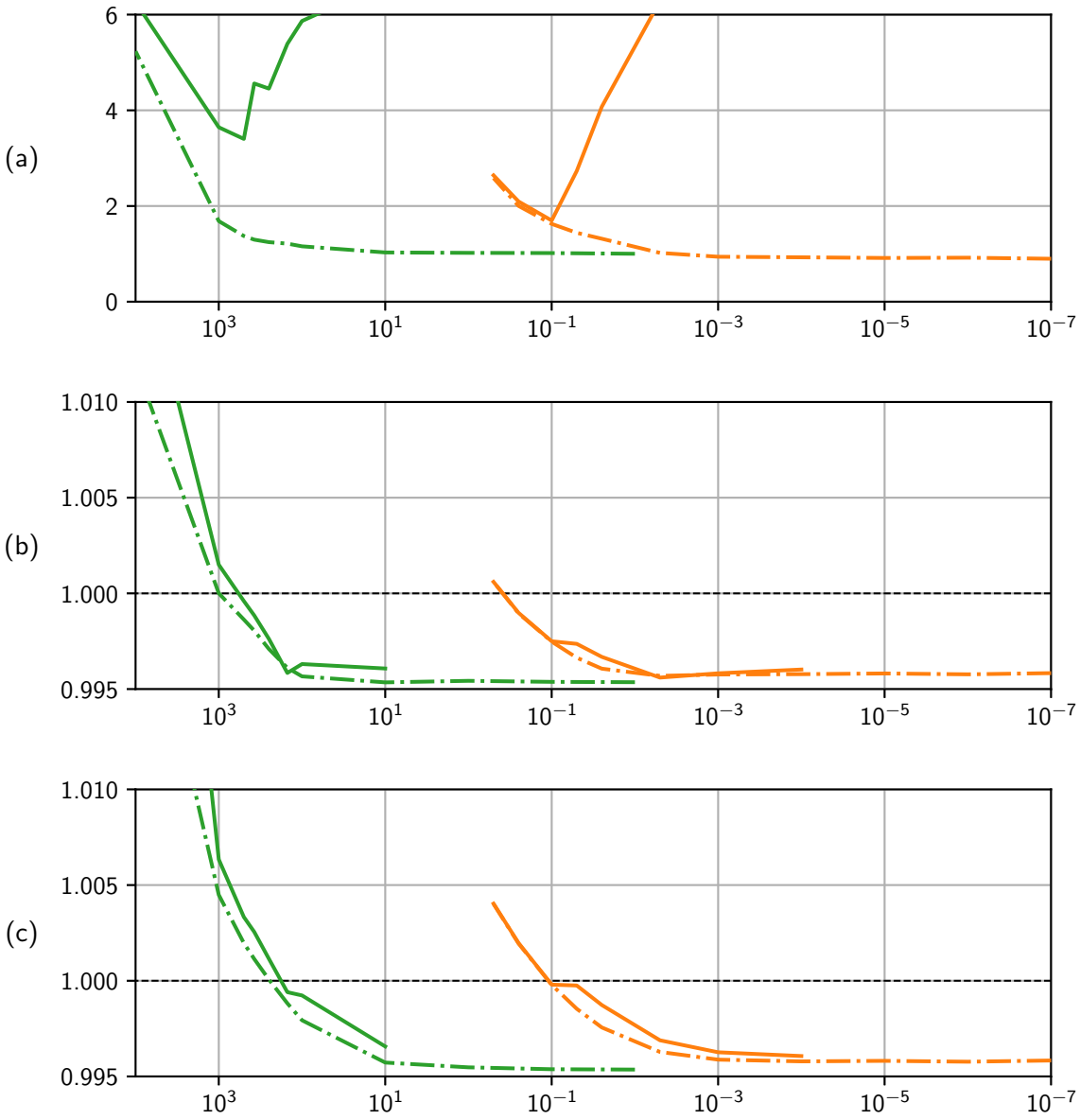


Figure 3.14: Mean relative error (a), relative data misfit (b) and relative total misfit (c) for several values of the regularization parameter λ and for both Tikhonov penalization (▬) or total variation penalization (▬), with (▬) or without (▬) preconditioning. The Tikhonov penalized misfit is combined with the isotropic inner product with $l_c = 5$ [m] while the total variation penalized misfit is combined with the anisotropic inner product with $l_c = 9$ [m].

both curves would be superimposed and similarly for two different models from the same flat minimum. This is not the case here: the preconditioned inversions yield a smaller total misfit, which shows that different minima are reached. The goal of the misfit penalization is to reduce the number of solutions: without penalization, several model parameters yield a small data misfit and there are multiple solutions. To discrimination between those solutions, a data misfit penalization is added and its strentgh should be increased such that only one solution emerges. In such a case, all the method converges to the same solution, whatever the preconditioner. This effect can be observed for the Tikhononv regularization with $\lambda > 0.1$: both

the preconditioned and un-preconditioned curves overlap. Preconditioning allows to diminish this minimal strength of the misfit penalization and allows to obtain a better reconstruction, as can be seen in Figure 3.15c or 3.15d. In Figure 3.14a, for the preconditioned methods, the error reaches a plateau for very small values of the regularization parameter λ, while it is expected to increase. This is because the inversion algorithm is hitting the 250 iteration limit. The data penalization misfit term is then inactive and the only regularization effect is early stopping with inner product preconditioning. Consequently, in the next paragraph, preconditioning alone is studied, because a data misfit penalization is not needed.

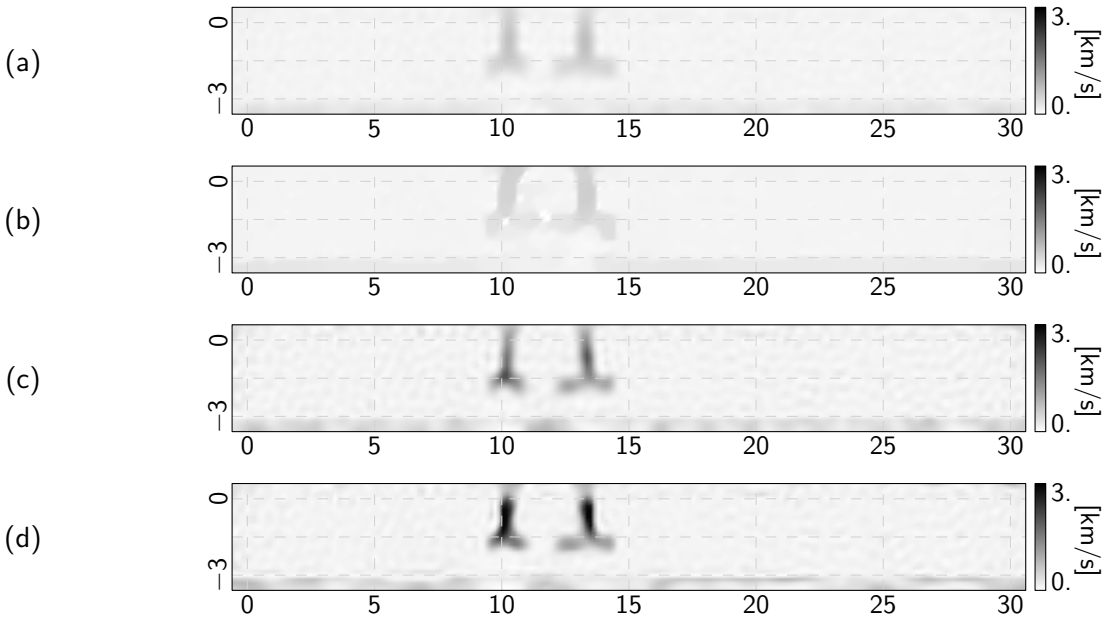


Figure 3.15: Near-surface concrete structures velocity inversion results using either a data misfit penalization regularization: Tikhonov (a), total variation (b) or inner product preconditioning regularization: isotropic (c), anisotropic (d). Stopping iterations and regularization parameters are chosen as the best in the sense of the mean relative error. Note that the upper color scale limit is only 3 [km/s].

	TIK	TV	ISO-IP	ANISO-IP
Minimal error	1.7	3.4	0.91	0.86

Table 3.4: Best mean relative error for different regularization strategies. Tikhonov (TIK) and total variation (TV) additive penalization. Isotropic (ISO-IP) and anisotropic (ANISO-IP) inner product preconditioning.

Inner product preconditioning In this case study, addiditive data misfit penalization methods need preconditioning to converge to a physically meaningful model. The opposite is not true however, as inner product preconditioned methods reach accurate reconstructions, even without misfit penalization. In the previous paragraph, the regularization parameter α

was fixed. In this paragraph, the influence of this choice is again discussed. Both preconditioned inversions are performed for different characteristic length, from $l_c = 1$ [m] which is the smallest propagated wavelength in the background to $l_c = 11$ [m]. For the edge-enhancing inner product, the scale structure ρ is chosen as half the wavelength while the edge threshold β is chosen such that the weak bottom reflector in the initial model is identified as an edge. The data and model convergence curves are given in Figure 3.17a-3.18a and Figure 3.17b-3.18b, while the L-curves, *i.e.* the data misfit plotted against the inner product penalty (3.48), are given in Figure 3.17c-3.18c. The error at the discrepancy limit and at the best iteration are summarized in Table 3.5-3.6. A comparison of the data and model convergence curves for the best smoothing length for each inner product is also given in Figure 3.16a and Figure 3.16b.

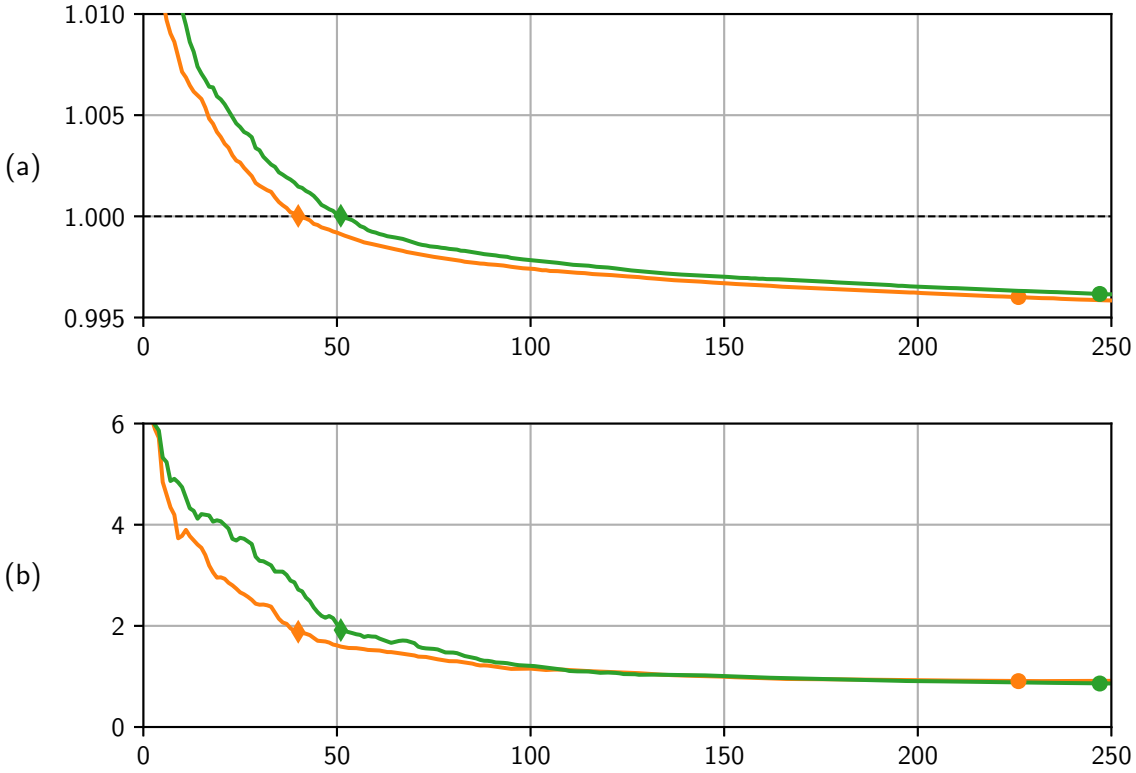


Figure 3.16: Relative data misfit (a) and mean relative error (b) for all the successive iterations, for an isotropic ($l_c = 5$ [m])(▬) or an anisotropic ($l_c = 9$ [m])(▬) inner product. Markers indicate the best iterations in the sense of the discrepancy principle (♦) and in the sense of the minimal error (●).

Similarly to the first case study, the convergence speed in the data space decreases when increasing the strength α of the inner product penalty, as expected. However, the model space convergence does not follow the same trend. For low characteristic lengths around the smallest propagated wavelength, the inverted model parameters demonstrate a large error, and the reconstructions are actually again close to the un-regularized inversions from Chapter 2 (*e.g.* Figure 2.11c). Increasing slightly the smoothing lengths to $l_c = 3$ [m] and above,

nevertheless gives very satisfactory results. The best result is again obtained with the edge-enhancing inner product with $l_c = 9$ [km] (*cf.* Figure 3.15d). The reconstruction for the isotropic inner product with $l_c = 5$ [m] is also very good, but the amplitude of the velocity anomalies are lower and further away from the ground-truth value. By tilting its smoothing direction, edge-enhancing preconditioning allows to better reconstruct sharp features. At a given smoothing length, its convergence speed in the data space is also slightly higher, because the discrepancy limit is reached faster, but the corresponding error is also larger than the isotropic inner product. The advantage of the edge-enhancing inner product really appears in the end of the inversion, as can be seen in Figue 3.16b. By design, the edge-enhancing inner product is built to image linear sharp structures. It is not well suited for corners, with two strong principal directions of variations instead of one. The diffusion tensor could however be adapted to accurately represent corners [114, 195]. Selecting the regularization parameter α is easier in this case, because the reconstructions with a too low smoothing lengths, *e.g.* Figure 2.11c from Chapter 2, are meaningless. The L-curves, which are given in Figure 3.17c-3.18c can help make this choice more formally. For a given level of data misfit, the smoothing length $l_c = 1$ and 3 [m] give a higher model penalty, which mean they should be discarded for a higher value.

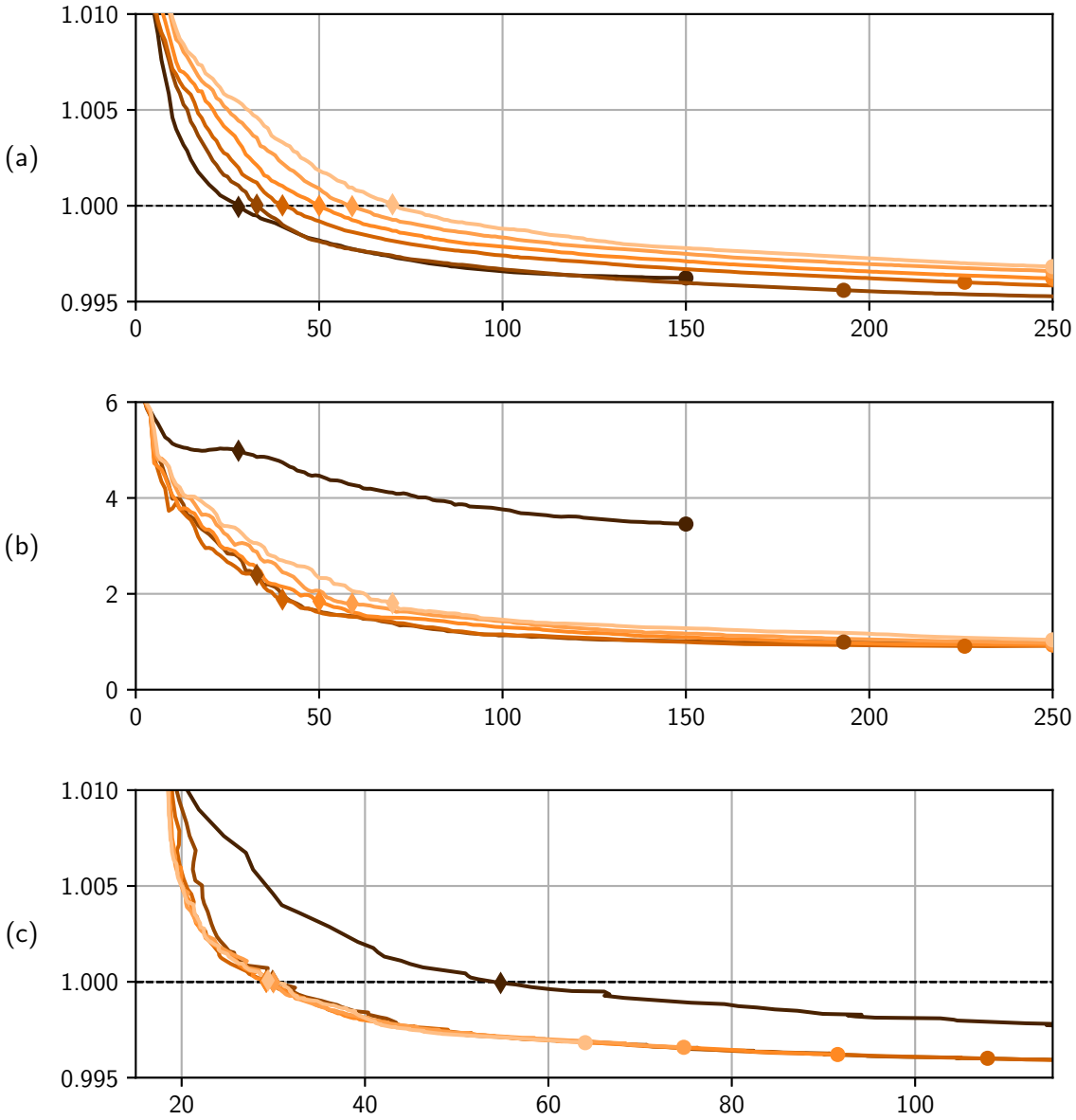


Figure 3.17: Mean relative error against the iteration number (a). Relative misfit against the iteration number (a) and against the inner product penalty (c). The isotropic inner product is used. Markers indicate the best iterations in the sense of the discrepancy principle (♦) and in the sense of the minimal error (●). l_c = 1.(▬), 3.(▬), 5.(▬), 7.(▬), 9.(▬), 11.(▬) [m].

l_c	1.0	3.	5.	7.	9.	11.
Minimal error	3.45	0.99	0.91	0.93	0.97	1.04
iteration #	150	193	226	250	250	250
Discrepancy	4.98	2.40	1.88	1.84	1.8	1.8
iteration #	28	33	40	50	59	70

Table 3.5: Mean relative error at the best iterations in the sense of the discrepancy principle and in the sense of the minimal error for inversions with a isotropic inner product with different smoothing lengths.

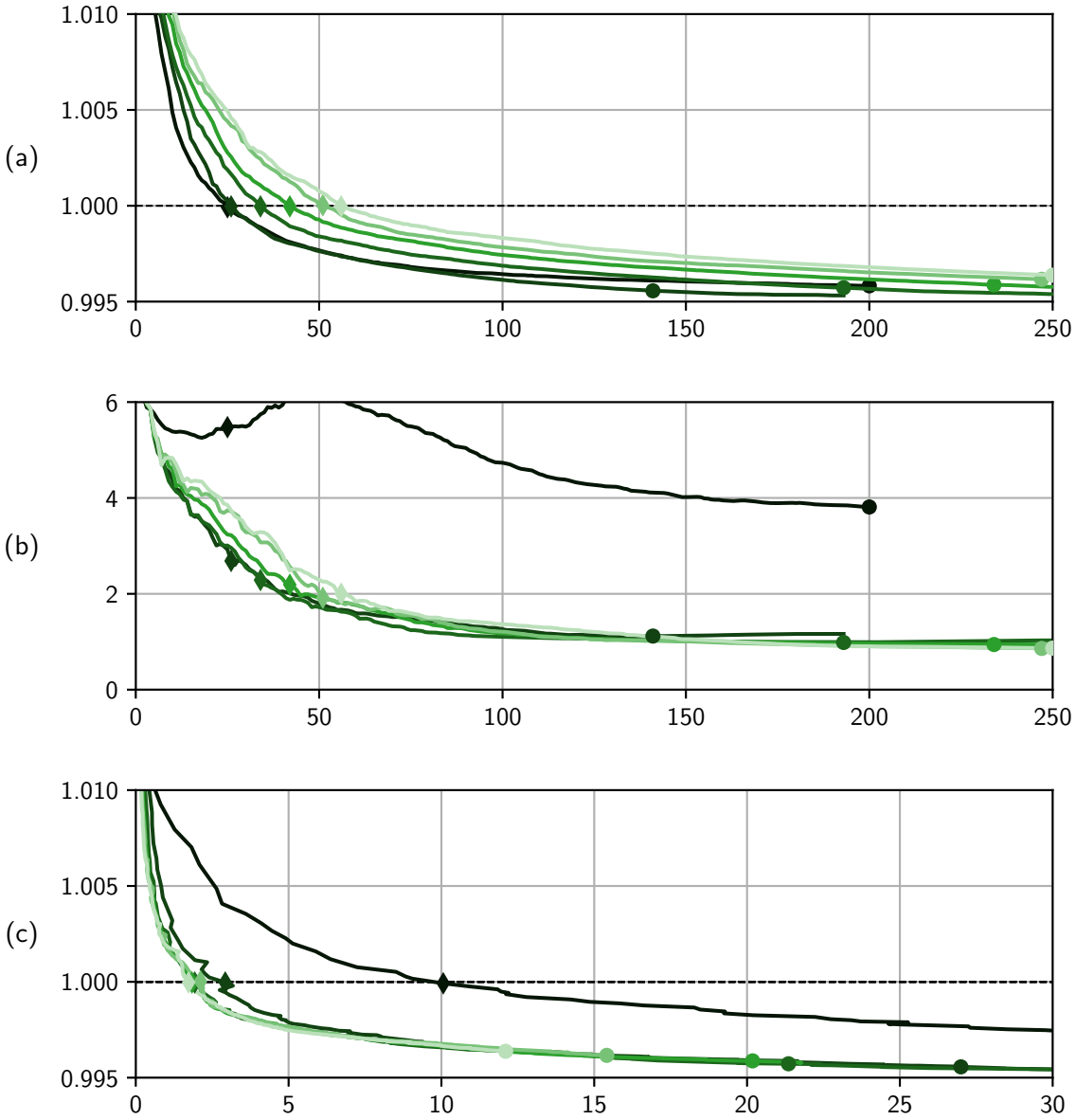


Figure 3.18: Mean relative error against the iteration number (a). Relative misfit against the iteration number (a) and against the inner product penalty (c). The anisotropic inner product is used. Markers indicate the best iterations in the sense of the discrepancy principle (♦) and in the sense of the minimal error (•). l_c = 1.(▬), 3.(▬), 5.(▬), 7.(▬), 9.(▬), 11.(▬) [m].

l_c	1.0	3.0	5.0	7.0	9.0	11.0
Minimal error	3.81	1.12	0.98	0.94	0.86	0.87
iteration #	200	141	193	234	247	250
Discrepancy	5.48	2.67	2.29	2.19	1.91	2.0
iteration #	25	26	34	42	51	56

Table 3.6: Mean relative error at the best iterations in the sense of the discrepancy principle and in the sense of the minimal error for inversions with an anisotropic inner product with different smoothing lengths.

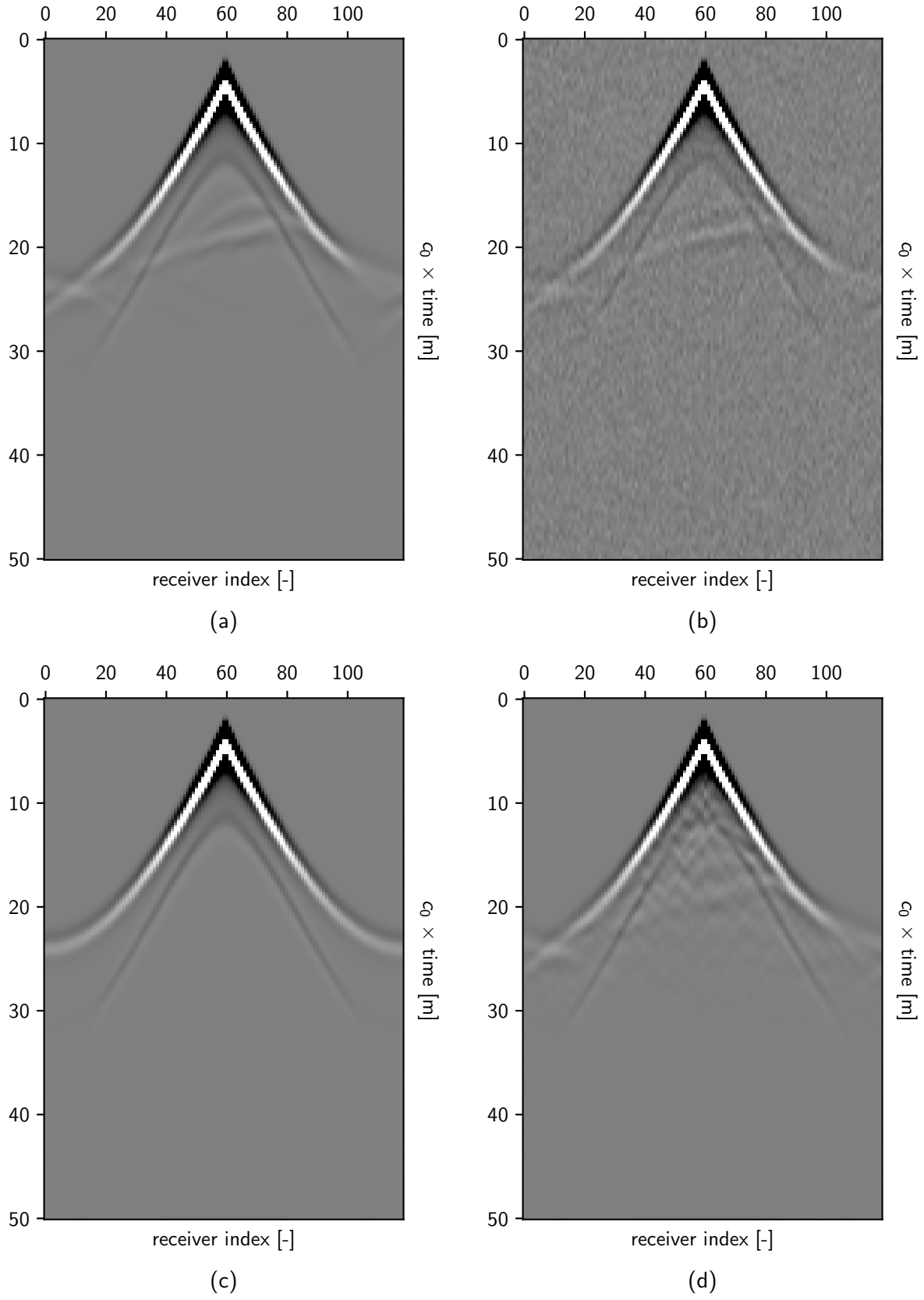


Figure 3.19: Time domain data for a single source, computed using a Newmark time-stepping scheme ($\gamma = 0.5$, $\beta = 0.25$, $c_0\Delta t = 10$ [mm], $c_0 T = 100$ [m]) for different conductivity and permittivity distributions. Ground-truth (a). Ground-truth distribution with additive Gaussian noise (SNR = 64 [-]) (b). Initial distribution (empty background) (c). Final distribution (*cf.* Figure 3.22g-3.22h) (d).

3.3.3 Case study 3: Dissipative crosses

Time domain data are given in Figure 3.19a. For this third case study, times are measured in meters, with the speed of light c_0 as the conversion factor, in the same way the pulsation ω was substituted by the wavenumber k in Chapter 2. These data, generated with a Ricker excitation ($f_c = 100$ [MHz]), are computed over a long time window ($c_0 T = 100$ [m]) with small time steps ($c_0 \Delta t = 10$ [mm]) and with finite elements of order 2. Noisy data are given in Figure 3.19b, a band limited Gaussian noise is used ($B = 250$ [MHz], SNR $= 64$ [-]). Time data corresponding to the initial and a typical final model parameter are also given in 3.19c and 3.19d. For all the inversion, the inner product is chosen such that permittivity updates are favored over to conductivity updates, *i.e.* $\beta = 3$, conforming with the guidelines from Chapter 3. The weight $w(\boldsymbol{x})$ is here chosen to be a constant inspired from the magnitude of the diagonal of the initial Gauss-Newton hessian. Again, the spatially varying part coming from the logarithmic change of variable is discarded, at the opposite of Chapter 2. The iterative procedure is stopped after a maximum of 200 iterations. Errors that are displayed in this case study are integrated over a reduced area that contains both crosses, *i.e.* a circle of radius $r = 3$ [m] centered at the origin. The mean relative error is computed for both relative permittivity and conductivity reconstructions and the mean of these two values is used as a figure of merit for the multi-parameter inversion.

Additive penalization In the context of a double-parameter inversion, there is no longer a single regularization parameter, but two: λ_ϵ and λ_σ. The hyper-parameter selection is therefore cumbersome: here, a grid of 7×7 inversions is computed for both Tikhonov and total variation penalties. The mean error for this grid is given in Figure 3.20 and the best couples of regularization parameters are deduced from that plot. The associated reconstructions are given in Figure 3.22e-3.22f and Figure 3.22g-3.22h while the corresponding mean relative errors are summarized in Table 3.7. Similarly to the noise-free case, the reconstruction of permittivities is more reliable than that of conductivities: cross shapes are better retrieved and the mean relative error is much smaller on the permittivity part. Also, the error is focused at the bottom cross, for which the values of both parameters are lower ($\epsilon_r = 1$ [-], $\sigma = 0.1$ [mS/m]), as can be seen from the error densities depicted in Figure 3.23a-3.23b and Figure 3.23c-3.23d. In particular, the conductivity of this bottom cross is very small and the data are thus not very sensitive to its value, hence the large error. Comparing both penalization methods, it appears that the total variation penalty term yields higher noise artifacts, particularly in the conductivity image. One explanation for this is the poor stability of the derivative in constant regions, where $|\mathbf{grad}\,(m)| \approx 0$, because the threshold that is used to prevent division by zero is only slightly above zero and it is not related to any model parameter jump. A possible solution is thus to introduce a physical edge threshold parameter in the definition of the total variation penalization, in the same spirit as the edge-enhancing inner product. Finally, the

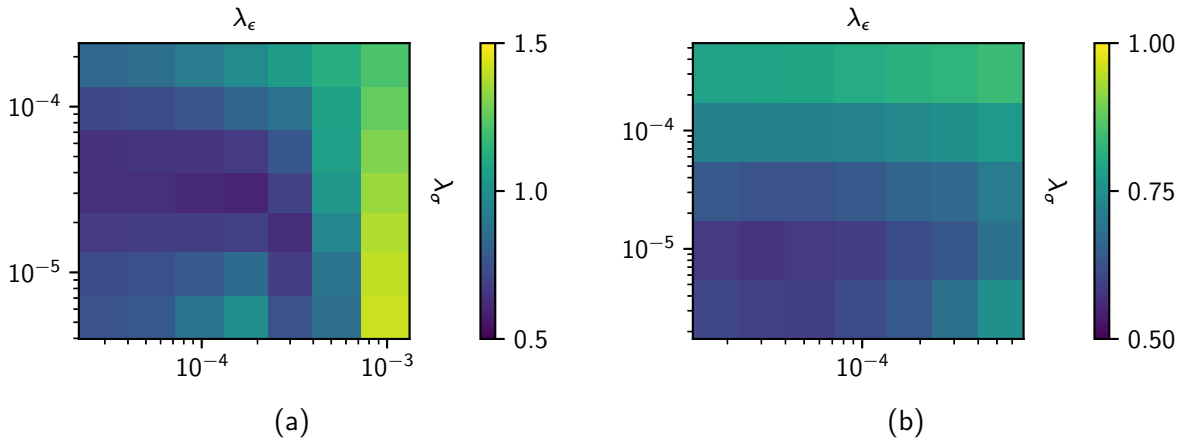


Figure 3.20: Mean relative error for several values of the regularization parameters λ_ϵ and λ_σ, for both Tikhonov penalization (a) or total variation penalization (b).

	TIK	TV	ISO-IP	ANISO-IP
Permittivity error	0.19	0.19	0.18	0.18
Conductivity error	0.96	1.01	1.0	0.9
Mean error	0.57	0.6	0.59	0.54

Table 3.7: Mean relative error for different regularization strategies. Tikhonov (TIK) and total variation (TV) additive penalization. Isotropic (ISO-IP) and anisotropic (ANISO-IP) inner product preconditioning. Regularization parameters for the penalization methods are chosen as the best in the sense of this error. Characteristic lengths for the isotropic and anisotropic inner products are chosen *a priori* ($l_c = 2.25$ [m] and $l_c = 3.75$ [m] respectively) and the iterative process is stopped after 100 iterations.

error history, *i.e.* the convergence in the model space, is given in Figure 3.21 as dashed lines for these two couples of best regularization parameters. The error increases for the first few iterations, because the model parameters are not constrained. It then starts to decrease after around 30 iterations but only drops below the initial error level after around 60 iterations. The situation is however better with inner preconditioned methods, as is explained in the following paragraph.

Inner product preconditioning To highlight the ease to select the regularization hyper-parameter α for the inner product preconditioned methods, only two characteristic lengths have been tried: $l_c = 2.25$ [m] and $l_c = 3.75$ [m]. The first one is of the order of the cross size and the second is slightly bigger. Moreover, the same smoothing strength is applied to both parameters, *i.e.* there is a single α. For the isotropic inner product, the best lengths is found to be $l_c = 2.25$ [m] while for the anisotropic and adaptative inner product, the best one is the largest ,*i.e.* $l_c = 3.75$ [m], in accordance with the conclusions of the first two

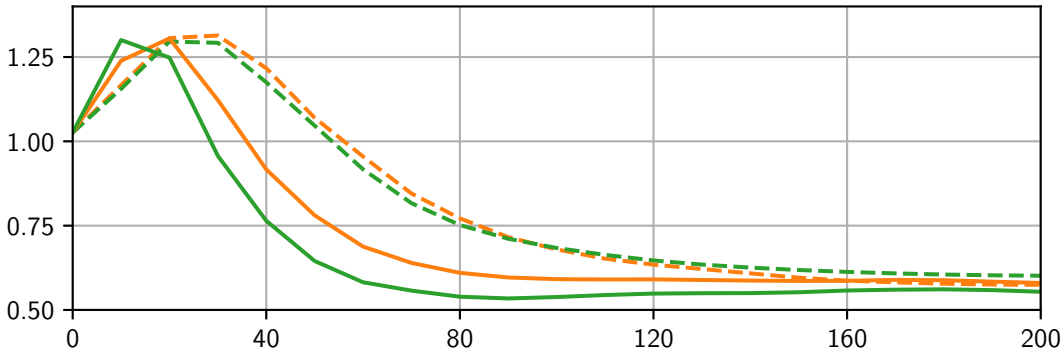


Figure 3.21: Mean relative error against the iteration number for different regularization strategies. Tikhonov (▪▪) and total variation (▪▪) additive penalization. Isotropic (▬) and anisotropic (▬) inner product preconditioning. Regularization parameters for the penalization methods are chosen as the best in the sense of this error. Characteristic lengths for the isotropic and anisotropic inner products are chosen *a priori* ($l_c = 2.25$ [m] and $l_c = 3.75$ [m] respectively)

case studies: the isotropic inner product is better with smaller smoothing lengths because it is not allowed to tilt its smoothing direction. In the context of this double parameter inversion, there are additional hyper-parameters to settle for the construction of the diffusion tensors. In particular, it is here chosen to use a shared structure tensor, as explained in the introduction, because the structure of the permittivity and the conductivity distribution is the same. In Chapter 2, it was shown that conductivity is less constrained by the data and that delaying the conductivity updates is beneficial for the inversion. This is also why the scaling factor between the two parameter classes is chosen bigger than one for this case study, *i.e.* $\beta = 3$. Following these observations, it is therefore chosen here to take the diffusion tensors to be based solely on the current permittivity reconstruction, *i.e.* $q_\epsilon = 1$ and $q_\sigma = 0$: firstly because during the iterative process, permittivity is reconstructed before conductivity and secondly because permittivity reconstruction is more reliable. Consequently, the structure is withdrew from the permittivity image and injected into the conductivity image through the diffusion tensor $\boldsymbol{D}$. The error history for both methods is given in Figure 3.21. As expected, the inversion method combined with the egde-enhancing inner product provides better results: the minimal error is smaller and actually, the error is smaller at any iteration. The improvement mainly appears for the conductivity reconstruction, as can be seen from Table 3.7, which gives the error separated for both the permittivity and conductivity parts, at the 100^{th} iteration. For this case study, the improvement over additive penalization methods is also huge: the minimal error is smaller and even more importantly, at the early stage of the inversion, there is a gap between both methods. After 40 iterations the error difference nearly reaches 50 % and it is still at 25 % after 80 iterations. Again, this is because regularization through preconditioning is effective from the beginning of the inversion procedure. Inversion results for both preconditioned methods are given in Figure 3.22e-3.22f and Figure 3.22g-3.22h.

Permittivity reconstructions are very similar and actually even slightly better in the direct vicinity of the crosses with the isotropic variant, as can also be seen in Figure 3.23e-3.23f and Figure 3.23g-3.23h. Indeed, at the inner corners of the crosses, the gradient is skewed at 45 degrees w.r.t. both the x-axis and the y-axis. In between two branches of a cross, near these inner corners, the diffusion tensor therefore favors diamond shapes and it results in a smoothing of the inner corners. This effect is particularly visible on the bottom cross. To avoid such an effect, the inner product penalization and specifically the diffusion tensor should be slightly more involved and in particular there should be no smoothing at corners, where its two principal directions have nearly the same eigenvalues [14, 65, 114, 195]. As far as the conductivity is concerned however, the bottom cross is better imaged with the edge-enhancing inner product but the rounding of inner corners still appears. The center of the cross and the amplitudes are better retrieved, because the structure is inherited from the permittivity distribution, which is of higher quality. Moreover, the amplitude of the noise artifacts around the crosses is smaller, because the smoothing in these regions is bigger than the isotropic inner product. This supplementary information brought through the diffusion tensor in the inner product allows a faster and more reliable conductivity reconstruction and overall makes the edge-enhancing inner product preconditioned methods a better approach.

3.4 Conclusion

In this chapter, inner product regularization methods are studied and compared to two other widely used regularization methods. Firstly, an overview of some sate-of-the-art regularization methods was given, followed by a reminder of two widely used additive misfit penalization methods, *i.e.* Tikhonov and total variation penalties. Then the concept of regularization through the inner product was introduced and analyzed through some toy problem examples. Two regularizing inner products were proposed: a smoothness-promoting inner product and an edge-enhancing inner product. While the former is very straightforward to define, the latter is slightly more subtle and is discussed in more detail. In order to highlight the advantages of inner product preconditioning, the three reference numerical case studies are considered, each of them demonstrating one or more benefits of this new approach, particularly when using the edge-enhancing variant. The first case study demonstrates that the edge-enhancing preconditioning is faster and yields more accurate results than the isotropic variant, because it is specifically designed to image linear structures. Conventional additive misfit regularization methods also work well on this first case study, because the initial model is close to the ground-truth; the situation might however be different with a less ideal starting model. The second case study illustrates the benefits of enforcing regularization from the beginning of the iterative process, as it is typically the case with preconditioning methods. Indeed, the local inversion process struggled to converge in the model space when regularization only

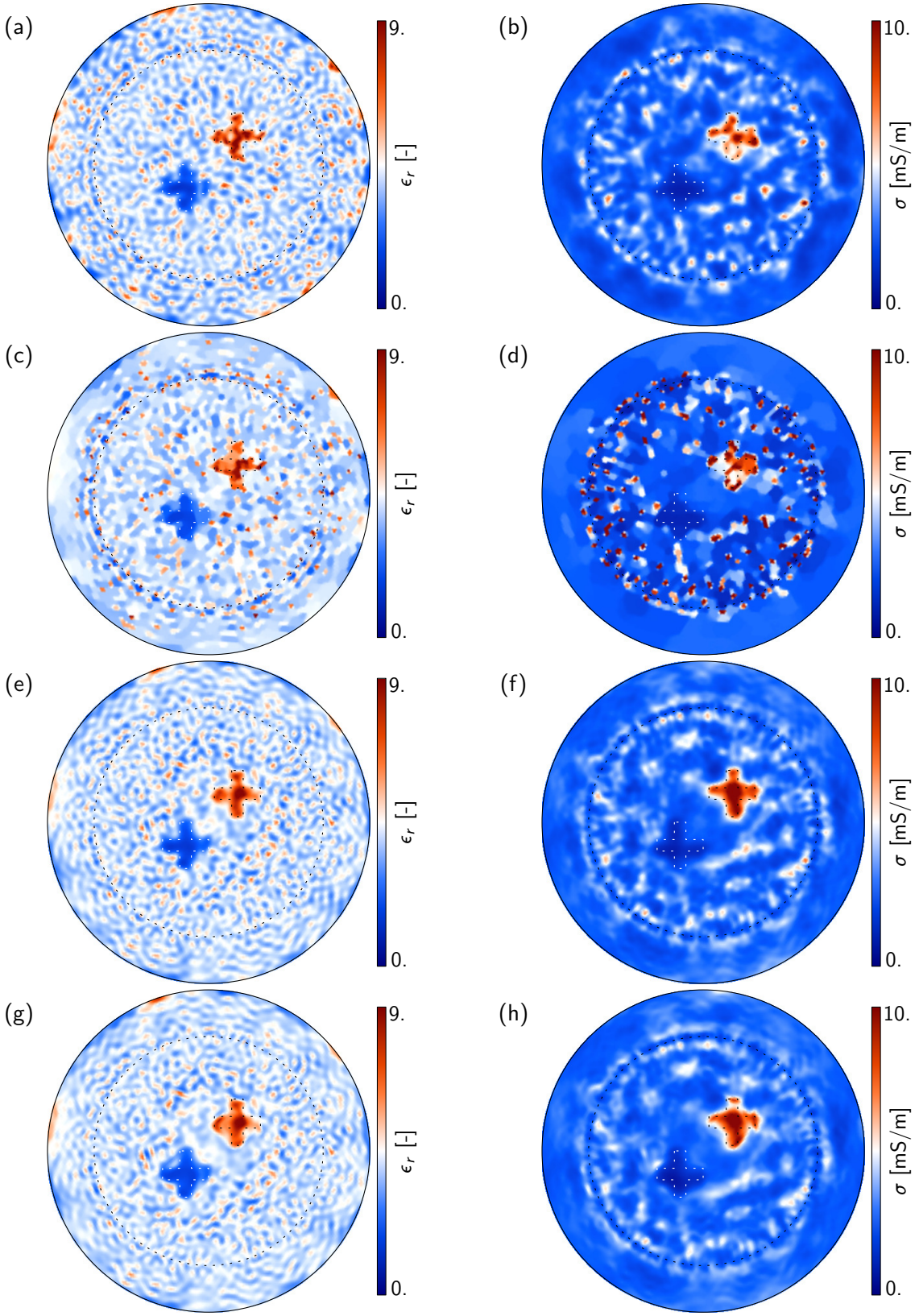


Figure 3.22: Inversion results using either a data misfit penalization: Tikhonov (a,b), total variation (c,d) or inner product preconditioning: isotropic (e,f), anisotropic (g,h). Regularization parameters for the penalization methods are chosen as the best in the sense of the mean relative error while characteristic lengths for the isotropic and anisotropic inner products are chosen *a priori*: $l_c = 2.25$ [m] and $l_c = 3.75$ [m] respectively. Left and right columns are respectively permittivity and conductivity reconstructions

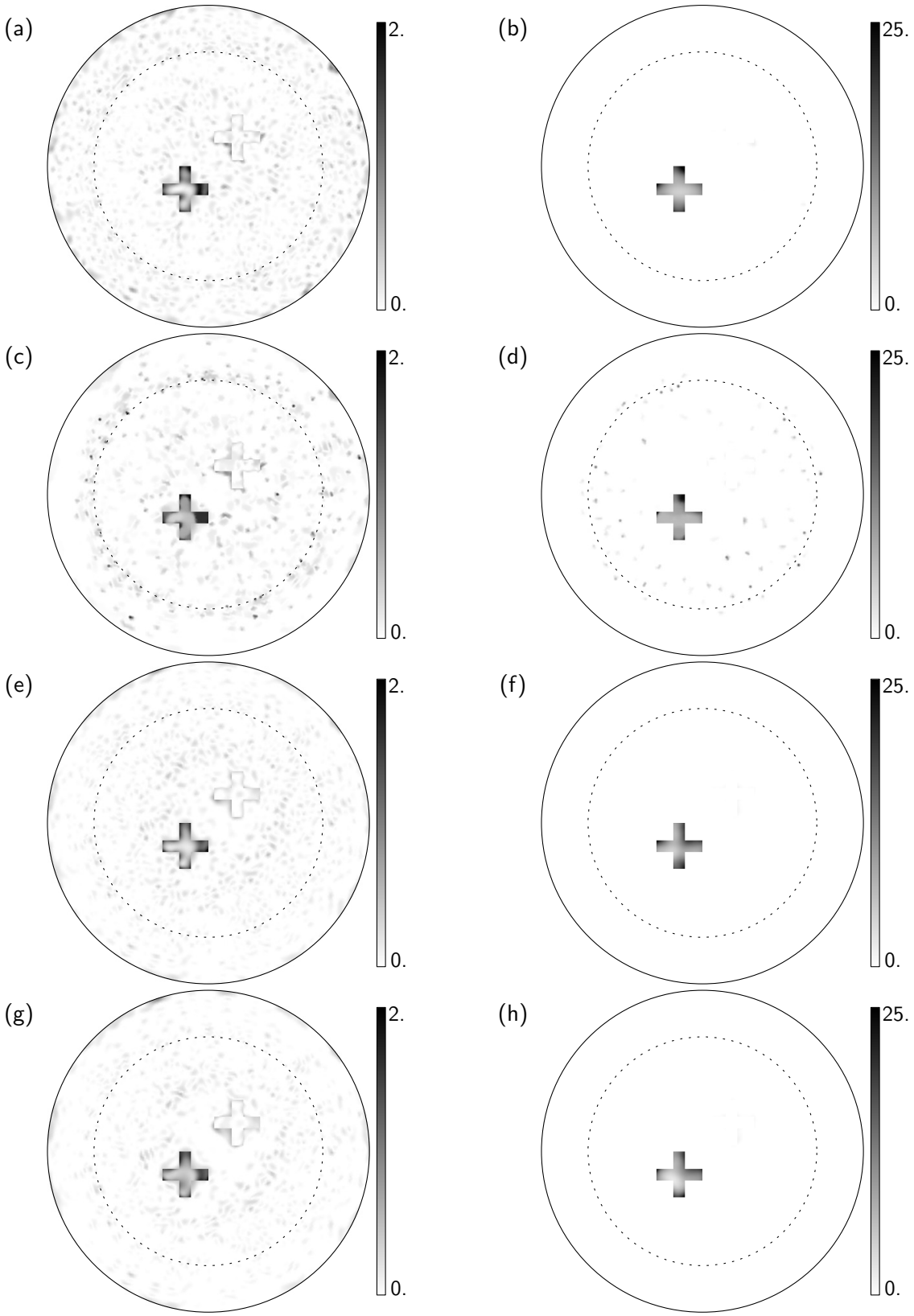


Figure 3.23: Relative error densities using either a data misfit penalization: Tikhonov (a,b), total variation (c,d) or inner product preconditioning: isotropic (e,f), anisotropic (g,h). Regularization parameters for the penalization methods are chosen as the best in the sense of the mean relative error while characteristic lengths for the isotropic and anisotropic inner products are chosen *a priori*: $l_c = 2.25$ [m] and $l_c = 3.75$ [m] respectively. Left and right columns are respectively related to permittivity and conductivity.

appears at the end of the inversion. Moreover, the edge-enhancing method was again slightly better, as it was able to reconstruct sharper variations in the model parameter. Finally, the third case study showed again the advantage of including some prior knowledge from the beginning, as preconditioned methods converged faster in the model space, *i.e.* the model error decreased faster, than additive penalization methods. Moreover, this double-parameter inversion gave some insights on the new possibilities offered by inner product preconditioning, such as softly coupling the geometrical structure both model parameters classes. Injecting information from a highly constrained parameter to an another less constrained parameter actually allowed to obtained better reconstructions. Importantly, during the discussion of these three case studies, it was shown how the inner product hyper-parameters are easy to select, in particular compared to conventional Tikhonov and total variation methods. In summary, regularization through edge-enhancing inner product is interesting for three main reasons. Firstly, its hyper-parameters are related to physical quantities, thus easier to choose. Moreover, the inversion results do not demonstrate a high sensitivity to these choices. Secondly, enforcing a regularization effect from the early stage of the inversion can allow or accelerate model error convergence. Thirdly, the edge-enhancing diffusion tensor is particularly suited to reconstruct linear structures: it gives a more accurate reconstruction of model parameter jumps, that are common in many natural media. Moreover this tensor can include, without effort, a soft geometrical coupling between several parameter classes. This inner product is however probably not the best one that could be defined: over-fitting the data is still observed in every case study. One straightforward improvement is to provide a specific treatment for corners, but much more substantial modifications, for example that do not take root in the diffusion filtering community, could also be considered to further enforce piece-wise linear reconstruction, or any other relevant model properties.

Conclusion

Full waveform inversion is a high-resolution imaging technique aiming to extract physical parameters of a medium, sampled by propagating waves, and which is based on matching the entire recorded data set with a numerically simulated data set, obtained through a mathematical model of the underlying wave propagation physics. The leading idea of full waveform inversion is a brute force approach to the imaging problem. For a given image, a synthetic data set can be generated numerically and the fundamental question of full waveform inversion then arises: *What image reproduces the recorded data set with the highest fidelity?* The numerical procedure generating these synthetic data sets, which specifically takes into account the full propagation physics, is the distinguishing feature that provides an unprecedented theoretical high-resolution to the method. This resolution increase has also been observed in practice, in several application fields such as marine and land seismic exploration, ground-penetrating radar, ultrasound medical imaging and non-destructive testing, where full waveform inversion is slowly becoming the industry standard. The interest in full waveform inversion is growing, because its major drawback, its numerical cost, is decreasing together with the ever-increasing available computational power. Research in full waveform inversion is therefore a pivotal stake in imaging science, as solving its challenges will spread its industrial integration and push its performance even further. These challenges are manifold. The greatest one remains the numerical cost, even though computational capabilities are growing. Constructing efficient algorithms, typically for solving the wave propagation problems and the minimization algorithms, remains a key issue. Then, with the great resolution power comes a great sensitivity to perturbations: noisy data, wave modeling errors or a very wrong initial image guess can lead to the failure of the procedure. Much effort must still be done to make full waveform inversion more robust and to automate the reconstruction process as much as possible, such that no human intervention or expertise is required during the imaging process. Finally, in the case a relevant image can be obtained successfully, the confidence that can be placed in that particular image naturally arise as the next concern to tackle: it is well-known that the image solution is not unique, but some features might be shared by all the possible solutions.

As mentioned in the introduction, the full waveform inversion workflow can be split into three parts or three spaces: the data space, the wave field space and the model space. In summary,

data space developments mainly focus on what is meant by *high fidelity* match between two data sets, *i.e.* the distance between two data sets; wave field space contributions aim to accelerate the construction of wave propagation problem solutions, *i.e.* the data generation for a given image; while model space improvements build new image update strategies, which incorporate prior knowledge on the surveyed sample and physical constraints, while maintaining a fast convergence to the optimal solution. This thesis was part of this last perspective. In a single sentence, it can be summarized as the study of the positive effects of appropriate metric modifications for the minimization problem arising in full waveform inversion. Throughout this thesis, the proposed ideas have systematically been tested on three numerical case studies, which were chosen to be representative of some typical challenges encountered in full waveform inversion: incomplete illumination, strong contrasts, sharp interfaces, multiple reflections and cross-talk among others.

Summary of main contributions

In Chapter 1, a general framework for the calculation of derivatives w.r.t. topological model parameters was introduced. This formalism included full waveform inversion, where the performance functional aims to minimize the discrepancy between observed data and simulated data, but was not limited to it. In particular, light was shed on the role of boundaries because in numerous applications, the wave propagation problem contains model parameters appearing in a boundary condition. Specifically, a unified derivation of the adjoint state method was presented, for performance functionals composed of both a volume and a boundary part and with model parameters living in the bulk and/or on the boundary. The central role of the inner product choice was also discussed. An exhaustive list of formulas was derived for the three most widely used wave propagation models, *i.e.* the Helmholtz equation, the Maxwell's equations and the Navier equation. Testing problems have been conducted systematically to demonstrate the validity of the formulas and their numerical implementations. At the end of that first chapter, an in-depth knowledge of gradients and the Hessian operators computation had been acquired, paving the way for their use in local optimization algorithm in the following chapter.

In Chapter 2, a large ensemble of local optimization algorithms were implemented and compared using the derivatives developed in the first chapter. Local optimization algorithms were introduced as the combination of three fundamental building blocks: the descent direction, the inner product and the globalization strategy. The three most widely used descent directions were investigated: the steepest direction, the *l*-BFGS direction and the (Gauss-)Newton direction; preconditioned through different metric choices, either to compensate for partial illumination, to smooth the updates, or to restore balance between the parameter classes in multi-parameter inversions; and lengthened with either a line search or a trust-

region procedure. Preconditioning can be presented from different viewpoints. In this thesis, the preconditioner wes seen as a consequence of metric modifications. This specific link was detailed and illustrated in this chapter. This choice of viewpoint is not innocuous: the metric also appears in the trust region constraint. Modifying this metric thus has a double effect, which are consistent with each other thanks to these definitions. In addition, viewing the preconditioner from the inner product aspect gave an alternative perspective, which helped in their design. Any of these ingredients taken separately is state-of-the-art, even though trust-region methods for full waveform inversion have hardly ever been explored before and similarly for Newton methods, which are often believed to be too computationally expensive. The novelties of this chapter were rather the unified presentation of the concepts around the inner product modification, the fair methodology of comparison and the extensive applications to full waveform inversion. The comparison of all the different combinations was a major original contributions of this thesis manuscript. Based on the three representative case studies, appropriate preconditioning was shown to dramatically increase the convergence speed, independently of any other choices. Moreover, it was shown that the (Gauss)-Newton should better be combined with a trust-region algorithm. With a trust-region globalization, the Newton methods surpassed its Gauss-Newton variants in many situations, because some of its drawbacks were removed. In the case of a double parameter inversion, (Gauss-)Newton methods nearly had the same convergence speed than the *l*-BFGS method, which overall was still the fastest descent direction.

In Chapter 3, a more subtle inner product was proposed, because the second chapter highlighted the pivotal role of preconditioning. The design of this new metric was inspired from an edge-enhancing operator frequently used in image processing. The focus of this third chapter was the robustness to noise in the observed data, *i.e.* the intrinsic regularization properties of the proposed optimization strategies. This study began with a comparison with conventional regularization techniques, namely the Tikhonov gradient penalization and the total variation penalization. During the presentation, the effect of inner product modifications is again reviewed, but this time under the angle of a metric penalization, because it enabled a direct comparison with conventional regularization methods: the penalization is in the metric rather than in the misfit. Inversions results related to the three reference case studies demonstrated that the image quality is similar or superior to the conventional methods when using the newly proposed filtering inner product, in addition to having multiple side advantages: firstly, preconditioning increased the size of the basin of attraction of the local minimum of interest, because the regularization effect appears from the beginning of the inversion; secondly, the regularization hyper-parameters were easier and more natural to choose; and thirdly it opened new possibilities such as a weak coupling between parameter classes, which allowed to share information between weakly constraint model parameters.

All the computational results contained in this thesis have been made possible through the

development of a flexible numerical framework, specific to the resolution of inversion problems based on wave propagation problems. This code constituted the main practical contribution of this thesis.

Proposal for further developments

While multiple aspects have been tackled throughout this thesis, there will always remain unexplored paths. The typical full waveform inversion procedure involves three quantities: data sets, wave fields and model parameters. In this thesis manuscript, the focus was on the model space, but there is also a lot of open problems in both the other spaces. A non-exhaustive list of potential improvements and research directions is given here below, from the most straightforward improvements of this thesis work to the most conceptual ones, for each part of full waveform inversion.

Model parameters space In Chapter 1, bulk sensitivities w.r.t. the model parameters of the most widely used wave equations were given. For any other equation, for example anisotropic wave equations [7, 24, 42, 56, 143], the formulas need to be updated and similarly for the boundary sensitivity formulas, which were given explicitly for some specific conditions only. Nevertheless, such extensions are straightforward because the procedure can be applied systematically. While volume topological derivatives were extensively used throughout the thesis, boundary derivatives have rather been left out. Applications taking profit of these boundary derivatives could be investigated, for example to determine the optimal equivalent impedance of a surface, in combination with an inverse homogeneization problem to create a micro-material with the appropriate equivalent macro-impedance [131, 151, 156].
In Chapter 2, truncated Newton methods were improved thanks to their combination with trust-region methods, but they were still slower than the *l*-BFGS method. A potential further improvement would thus to use the *l*-BFGS direction as the first guess of the inner iteration loop, instead of the gradient [118]. Moreover, it would be interesting to compare both methods on a case study gathering all the difficulties: multiple reflections, partial illumination and parameter cross-talk, as reducing the illumination contributes to enhance the ambiguity between parameter classes [72]. Alternatively, other approximate methods could be used to solve the Newton system. For example, instead of probing the Hessian operator in a series of directions generated by the conjugate gradient algorithm, an approximate Hessian operator could be built from a series of parameterized elementary operators, that are determined from the application of the Hessian in a small number of random directions [38]. Another possibility is to explore other quasi-Newton methods, *i.e.* methods from the same family than *l*-BFGS, which approaches the inverse Hessian in a low-rank form [105, 106]. The advantages of both these methods is not limited to potentially accelerate the converge speed: they also

allow for a model resolution analysis through the sampling of the posterior model distribution, which is nothing else that the inverse Hessian [21, 54]. Resolution analysis and uncertainty quantification are the natural concerns that arise once a particular solution is found. Future researches should ideally include the computational cost of such an analysis when comparing different reconstruction methods. Around a given minimum, this information can be extracted from the inverse Hessian through a randomized singular value decomposition [21], but its computational requirements are high. Instead, it could thus be interesting to construct a Hessian approximation during the inversion process which is easily probable [105], or to reduce the computational burden of the randomized singular value decomposition thanks to point-spread functions approximations [53, 208]. Another very recent approach, which does not require to probe the inverse Hessian, is to compute statistical estimates on a large ensemble of complete inversion results obtained from randomly generated initial guesses [179]. All the inversion runs are thus embarrassingly parallel, which constitutes a huge advantage on modern computer clusters.
In Chapter 3, the robustness of the regularization through preconditioning could be analyzed further, by repeating the analysis for different noise levels. In addition, it would be interesting to implement a diffusion tensor which behaves appropriately on corners and to compare line search and trust-region methods in the context of noisy data. Due to its good performance, the adaptative and anisotropic inner product could also be introduce in the misfit, as a conventional additive penalization method [14]. A probabilistic view of the metric modification would also be interesting to consider, as it can be linked to the *a priori* model co-variance operator [177]. Moreover, the proposed preconditioners could be improved, as the discrepancy limit is still crossed before the model error minimum is reached. Preconditioner based on the seislet transform could be considered when the model properties are known to be sparse in that basis [199]. If such a basis is unknown, then the preconditioner can be learned from available or synthetic reconstruction examples [91, 123, 183].

Data space Data sets used throughout this thesis are synthetic and an assessment of the proposed concepts on real data sets, contaminated with errors more complex than a white Gaussian noise, would be very instructive. For example, when inverting seismic data within the acoustic approximation, some elastic imprints on the data are inevitable and a heavy pre-processing must then be done, *e.g.* thanks to an elastic-to-acoustic converter, previously trained on acoustic-elastic couples of data sets [100, 190, 204]. The distance functional between the observed and simulated data sets has not been studied neither in this thesis, although its pivotal role in preventing cycle-skipping and thus in increasing full waveform inversion robustness. In the time domain, two approaches have demonstrated to be particularly efficient: the first is based on matching filters [192] and the second is based on optimal transport [115]. Transposing both these functionals in the frequency domain is doubtlessly

an interesting perspective. Nevertheless, cycle-skipping only appears when the low-frequency data content is missing: a straightforward idea is thus to extrapolate this low frequency content from the available frequency band [101, 175].

Wave field space For the last two chapters, only the simplest wave propagation physics, *i.e.* the Helmholtz equation, has been considered. The extension of inversions to more complex physics might also be needed when considered real data sets. Moreover, depending on the surveyed area, three dimensional images will also be required. For small two dimensional imaged regions, direct solvers, relying on a upper-lower factorization of the forward system, are affordable and are actually particularly useful in a context of multiple shots, because each excitation only require two triangular solves. For larger domains however, typically in three dimensions, this factorization, whose memory requirements are proportional to the fourth power of the matrix size, becomes prohibitively expensive. The usual alternative is then iterative methods. Unfortunately, systems arising form the discretization of wave scattering equations in the frequency domain are ill-conditioned and alternative methods must be employed [48]. Considering the particular structure of the full waveform inversion workflow, *i.e.* slowly evolving scattering problems solved for each outer optimization iteration, particular methods, that take benefits from this redundancy, should be investigated. Among these is the controllability method. In this approach, a time-periodic solution, which thus takes advantage of the good scalability properties of the time-domain solvers, is computed through an optimization algorithm. Successive wave propagation problems that appear during the outer optimizations, related to model inversion, are then progressively solved faster as the initial guesses for the wave field appearing in the inner optimization can be chosen based on the wave field history [176]. On the other hand, large scale wave scattering propagation problem in the frequency domain are often solved through domain decomposition methods: a hybrid direct-iterative family of methods which are parallel by design. These methods propose to split the resolution domain in several parts; then to solve on each subdomains the wave propagation problem by direct factorization; and finally to iterate between the different subdomains through a reduced system defined on the interfaces [81, 172]. When included in a full waveform inversion loop, these methods become computationally expensive, as a supplementary inner iterative process needs to be performed at every outer iteration. A promising idea are then *multi-step one-shot* methods, which iterate on the forward problem solution and on the inverse problem for model parameters in one same loop [15]. In particular, their combination with a waveform reconstruction formulation of full waveform inversion could be interesting. According to waveform reconstruction inversion, the forward wave propgation problem is not enforced exactly, but rather through a penalty in the misfit [98].

Appendix A

Adjoint operators

This appendix provides the derivation detail of the adjoint operators introduced in Chapter 1.

A.1 State space

A.1.1 Helmholtz I

The adjoint of the Gâteaux derivative w.r.t. u of the Helmholtz I's operator is

$$\partial_u^\dagger F = \operatorname{div}(\mathbf{grad}\,()) + \omega^2 \overline{s}^2 = \overline{\partial_u F} \tag{A.1}$$

while the boundary term is

$$\left[u^\dagger, u\right]_{\partial_u F} = \int_{\partial\Omega} \begin{pmatrix} \frac{\partial \overline{u}^\dagger}{\partial n} & \overline{u}^\dagger \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial u}{\partial n} \\ u \end{pmatrix} d\partial\Omega \tag{A.2}$$

where $\frac{\partial}{\partial n} := \hat{\boldsymbol{n}} \cdot \mathbf{grad}\,()$ denotes the normal derivative. Indeed

$$\left\langle u^\dagger, \{\partial_u F\}(u) \right\rangle_{L_2(\Omega)} = \int_\Omega \overline{u}^\dagger \operatorname{div}(\mathbf{grad}\,(u))\, d\Omega + \int_\Omega \overline{u}^\dagger \omega^2 s^2 u\, d\Omega \tag{A.3}$$

$$= \int_\Omega \operatorname{div}\left(\mathbf{grad}\left(\overline{u}^\dagger\right)\right) u\, d\Omega + \int_\Omega \overline{u}^\dagger \omega^2 s^2 u\, d\Omega + \int_{\partial\Omega} \left(\overline{u}^\dagger \frac{\partial u}{\partial n} - u \frac{\partial \overline{u}^\dagger}{\partial n}\right) d\partial\Omega \tag{A.4}$$

$$= \int_\Omega \overline{(\operatorname{div}(\mathbf{grad}\,(u^\dagger)) + \omega^2 \overline{s}^2 u^\dagger)} u\, d\Omega + \int_{\partial\Omega} \left(\overline{u}^\dagger \frac{\partial u}{\partial n} - u \frac{\partial \overline{u}^\dagger}{\partial n}\right) d\partial\Omega. \tag{A.5}$$

A.1.2 Helmholtz II

The adjoint of the Gâteaux derivative w.r.t. u of the Helmholtz II's operator is

$$\partial_u^\dagger F = \operatorname{div}(\overline{\nu}\,\mathbf{grad}\,()) + \omega^2\overline{K} = \overline{\partial_u F} \tag{A.6}$$

while the boundary term is

$$\left[u^\dagger, u\right]_{\partial_u F} = \int_{\partial\Omega} \begin{pmatrix} \nu\frac{\partial\overline{p}^\dagger}{\partial n} & \overline{p}^\dagger \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \nu\frac{\partial p}{\partial n} \\ p \end{pmatrix} d\partial\Omega \tag{A.7}$$

where $\frac{\partial}{\partial n} := \hat{\boldsymbol{n}} \cdot \mathbf{grad}\,()$ denotes the normal derivative. Indeed

$$\left\langle u^\dagger, \{\partial_u F\}(u)\right\rangle_{L_2(\Omega)} = \int_\Omega \overline{p}^\dagger \operatorname{div}(\nu\,\mathbf{grad}\,(p))\; d\Omega + \int_\Omega \overline{p}^\dagger \omega^2 K p\, d\Omega \tag{A.8}$$

$$= \int_\Omega \operatorname{div}\left(\nu\,\mathbf{grad}\left(\overline{p}^\dagger\right)\right) p\, d\Omega + \int_\Omega \overline{p}^\dagger \omega^2 K p\, d\Omega + \int_{\partial\Omega} \left(\overline{p}^\dagger\, \nu\frac{\partial p}{\partial n} - p\,\nu\frac{\partial\overline{p}^\dagger}{\partial n}\right) d\partial\Omega \tag{A.9}$$

$$= \int_\Omega \overline{\left(\operatorname{div}(\overline{\nu}\,\mathbf{grad}\,(p^\dagger)) + \omega^2\overline{K}p^\dagger\right)} p\, d\Omega + \int_{\partial\Omega} \left(\overline{p}^\dagger\, \nu\frac{\partial p}{\partial n} - p\,\nu\frac{\partial\overline{p}^\dagger}{\partial n}\right) d\partial\Omega. \tag{A.10}$$

A.1.3 Maxwell

The adjoint of the Gâteaux derivative w.r.t. u of the Maxwell operator is

$$\partial_u^\dagger F = \begin{pmatrix} i\omega\overline{\epsilon} & \mathbf{curl}\,() \\ \mathbf{curl}\,() & -i\omega\overline{\mu} \end{pmatrix} = \overline{\partial_u F}. \tag{A.11}$$

while the boundary term is

$$\left[u^\dagger, u\right]_{\partial_u F} = \int_{\partial\Omega} \begin{pmatrix} \gamma_t(\overline{\boldsymbol{h}}^\dagger) & \gamma_T(\overline{\boldsymbol{e}}^\dagger) \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \gamma_t(\boldsymbol{h}) \\ \gamma_T(\boldsymbol{e}) \end{pmatrix} d\partial\Omega \quad (\boldsymbol{e}\text{-formulation}) \tag{A.12}$$

$$= \int_{\partial\Omega} \begin{pmatrix} \gamma_t(\overline{\boldsymbol{e}}^\dagger) & \gamma_T(\overline{\boldsymbol{h}}^\dagger) \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \gamma_t(\boldsymbol{e}) \\ \gamma_T(\boldsymbol{h}) \end{pmatrix} d\partial\Omega \quad (\boldsymbol{h}\text{-formulation}) \tag{A.13}$$

where $\gamma_T() := -\hat{\boldsymbol{n}} \times (\hat{\boldsymbol{n}}\times)$ denotes the tangential component and $\gamma_t() := \hat{\boldsymbol{n}}\times$ denotes the orthogonal tangential component. Indeed

$$\left\langle u^\dagger, \{\partial_u F\}(u)\right\rangle_{L_2^3(\Omega)\times L_2^3(\Omega)}$$

$$= \int_\Omega \left[\overline{\boldsymbol{e}}^\dagger \cdot (\mathbf{curl}\,(\boldsymbol{h}) - i\omega\epsilon\boldsymbol{e})\right]\, d\Omega + \int_\Omega \left[\overline{\boldsymbol{h}}^\dagger \cdot (\mathbf{curl}\,(\boldsymbol{e}) + i\omega\mu\boldsymbol{h})\right]\, d\Omega \tag{A.14}$$

$$= \int_\Omega \left[-\mathrm{div}\left(\overline{\boldsymbol{e}}^\dagger \times \boldsymbol{h}\right) + \mathbf{curl}\left(\overline{\boldsymbol{e}}^\dagger\right) \cdot \boldsymbol{h} - i\omega\epsilon\overline{\boldsymbol{e}}^\dagger \cdot \boldsymbol{e}\right]\, d\Omega + \int_\Omega \left[-\mathrm{div}\left(\overline{\boldsymbol{h}}^\dagger \times \boldsymbol{e}\right) + \mathbf{curl}\left(\overline{\boldsymbol{h}}^\dagger\right) \cdot \boldsymbol{e} + i\omega\mu\overline{\boldsymbol{h}}^\dagger \cdot \boldsymbol{h}\right]\, d\Omega \tag{A.15}$$

$$= \int_\Omega \overline{\left(\mathbf{curl}\,(\boldsymbol{e}^\dagger) - i\omega\overline{\mu}\boldsymbol{h}^\dagger\right)} \cdot \boldsymbol{h}\, d\Omega + \int_\Omega \overline{\left(\mathbf{curl}\left(\boldsymbol{h}^\dagger\right) + i\omega\overline{\epsilon}\boldsymbol{e}^\dagger\right)} \cdot \boldsymbol{e}\, d\Omega - \int_{\partial\Omega} \left(\overline{\boldsymbol{e}}^\dagger \times \boldsymbol{h}\right) \cdot \hat{\boldsymbol{n}}\, d\partial\Omega - \int_{\partial\Omega} \left(\overline{\boldsymbol{h}}^\dagger \times \boldsymbol{e}\right) \cdot \hat{\boldsymbol{n}}\, d\partial\Omega \tag{A.16}$$

and

$$\left[u^\dagger, u\right]_{\partial_u F} = -\int_{\partial\Omega} \left(\overline{\boldsymbol{e}}^\dagger \times \boldsymbol{h}\right) \cdot \hat{\boldsymbol{n}}\, d\partial\Omega - \int_{\partial\Omega} \left(\overline{\boldsymbol{h}}^\dagger \times \boldsymbol{e}\right) \cdot \hat{\boldsymbol{n}}\, d\partial\Omega \tag{A.17}$$

$$= \int_{\partial\Omega} \overline{\boldsymbol{e}}^\dagger \cdot \gamma_t(\boldsymbol{h})\, d\partial\Omega - \int_{\partial\Omega} \gamma_t(\overline{\boldsymbol{h}}^\dagger) \cdot \boldsymbol{e}\, d\partial\Omega \tag{A.18}$$

$$= \int_{\partial\Omega} \gamma_T(\overline{\boldsymbol{e}}^\dagger) \cdot \gamma_t(\boldsymbol{h})\, d\partial\Omega - \int_{\partial\Omega} \gamma_t(\overline{\boldsymbol{h}}^\dagger) \cdot \gamma_T(\boldsymbol{e})\, d\partial\Omega \tag{A.19}$$

$$= \int_{\partial\Omega} \begin{pmatrix}\gamma_t(\overline{\boldsymbol{h}}^\dagger) & \gamma_T(\overline{\boldsymbol{e}}^\dagger)\end{pmatrix} \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} \begin{pmatrix}\gamma_t(\boldsymbol{h})\\ \gamma_T(\boldsymbol{e})\end{pmatrix}\, d\partial\Omega \tag{A.20}$$

or by symmetry

$$= \int_{\partial\Omega} \begin{pmatrix}\gamma_t(\overline{\boldsymbol{e}}^\dagger) & \gamma_T(\overline{\boldsymbol{h}}^\dagger)\end{pmatrix} \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} \begin{pmatrix}\gamma_t(\boldsymbol{e})\\ \gamma_T(\boldsymbol{h})\end{pmatrix}\, d\partial\Omega. \tag{A.21}$$

A.1.4 Navier

The adjoint of the Gâteaux derivative w.r.t. u of the Navier operator is

$$\partial_u^\dagger F = \mathrm{div}\,(\overline{\boldsymbol{\sigma}}()) + \omega^2\overline{\rho} = \overline{\partial_u F}. \tag{A.22}$$

while the boundary term is

$$\left[u^\dagger, u\right]_{\partial_u F} = \int_{\partial\Omega} \begin{pmatrix}\boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \hat{\boldsymbol{n}} & \gamma_n(\overline{\boldsymbol{u}}^\dagger)\end{pmatrix} \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} \begin{pmatrix}\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}}\\ \gamma_n(\boldsymbol{u})\end{pmatrix}\, d\partial\Omega + \int_{\partial\Omega} \begin{pmatrix}\boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \hat{\boldsymbol{n}} & \gamma_T(\overline{\boldsymbol{u}}^\dagger)\end{pmatrix} \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} \begin{pmatrix}\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}}\\ \gamma_T(\boldsymbol{u})\end{pmatrix}\, d\partial\Omega \tag{A.23}$$

where $\gamma_n() := \hat{\boldsymbol{n}}\,(\hat{\boldsymbol{n}}\cdot)$ denotes the normal component and $\gamma_T() := -\hat{\boldsymbol{n}} \times (\hat{\boldsymbol{n}}\times)$ denotes the tangential component. Indeed

$$\left\langle u^\dagger, \{\partial_u F\}(u)\right\rangle_{L_2^3(\Omega)} \tag{A.24}$$

$$= \int_\Omega \overline{\boldsymbol{u}}^\dagger \cdot \operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u}))\, d\Omega + \int_\Omega \omega^2 \rho \overline{\boldsymbol{u}}^\dagger \cdot \boldsymbol{u}\, d\Omega \tag{A.25}$$

$$= \int_\Omega \left[\operatorname{div}\left(\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \overline{\boldsymbol{u}}^\dagger\right) - \mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) : \boldsymbol{\sigma}(\boldsymbol{u})\right] d\Omega + \int_\Omega \omega^2 \rho \overline{\boldsymbol{u}}^\dagger \cdot \boldsymbol{u}\, d\Omega \tag{A.26}$$

$$= \int_\Omega \operatorname{div}\left(\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \overline{\boldsymbol{u}}^\dagger\right) d\Omega - \int_\Omega \mathbf{grad}(\boldsymbol{u}) : \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger)\, d\Omega + \int_\Omega \omega^2 \rho \overline{\boldsymbol{u}}^\dagger \cdot \boldsymbol{u}\, d\Omega \tag{A.27}$$

$$= \int_\Omega \operatorname{div}\left(\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \overline{\boldsymbol{u}}^\dagger - \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \boldsymbol{u}\right) d\Omega + \int_\Omega \operatorname{div}\left(\boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger)\right) \cdot \boldsymbol{u}\, d\Omega + \int_\Omega \omega^2 \rho \overline{\boldsymbol{u}}^\dagger \cdot \boldsymbol{u}\, d\Omega \tag{A.28}$$

$$= \int_\Omega \overline{\left(\operatorname{div}(\overline{\boldsymbol{\sigma}}(\boldsymbol{u}^\dagger)) + \omega^2 \overline{\rho} \boldsymbol{u}^\dagger\right)} \cdot \boldsymbol{u}\, d\Omega + \int_{\partial\Omega} \hat{\boldsymbol{n}} \cdot \left(\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \overline{\boldsymbol{u}}^\dagger - \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \boldsymbol{u}\right) d\partial\Omega \tag{A.29}$$

because the following identity holds

$$\mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) : \boldsymbol{\sigma}(\boldsymbol{u}) = \mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) : \left(\lambda \operatorname{div}(\boldsymbol{u})\, \boldsymbol{I} + \mu(\mathbf{grad}(\boldsymbol{u}) + \mathbf{grad}^T(\boldsymbol{u}))\right) \tag{A.30}$$

$$= \lambda \operatorname{div}\left(\overline{\boldsymbol{u}}^\dagger\right) \operatorname{div}(\boldsymbol{u}) + \mu \mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) : (\mathbf{grad}(\boldsymbol{u}) + \mathbf{grad}^T(\boldsymbol{u})) \tag{A.31}$$

$$= \lambda \operatorname{div}(\boldsymbol{u}) \operatorname{div}\left(\overline{\boldsymbol{u}}^\dagger\right) + \mu \mathbf{grad}(\boldsymbol{u}) : (\mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) + \mathbf{grad}^T\left(\overline{\boldsymbol{u}}^\dagger\right)) \tag{A.32}$$

$$= \mathbf{grad}(\boldsymbol{u}) : \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger). \tag{A.33}$$

The boundary term can also be expressed as

$$\left[u^\dagger, u\right]_{\partial_u F} = \int_{\partial\Omega} \overline{\boldsymbol{u}}^\dagger \cdot \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}} - \boldsymbol{u} \cdot \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \hat{\boldsymbol{n}}\, d\partial\Omega \tag{A.34}$$

$$= \int_{\partial\Omega} \left(\gamma_n(\overline{\boldsymbol{u}}^\dagger) + \gamma_T(\overline{\boldsymbol{u}}^\dagger)\right) \cdot \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}} - (\gamma_n(\boldsymbol{u}) + \gamma_T(\boldsymbol{u})) \cdot \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \hat{\boldsymbol{n}}\, d\partial\Omega \tag{A.35}$$

$$= \int_{\partial\Omega} \begin{pmatrix} \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \hat{\boldsymbol{n}} & \gamma_n(\overline{\boldsymbol{u}}^\dagger) \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}} \\ \gamma_n(\boldsymbol{u}) \end{pmatrix} d\partial\Omega \tag{A.36}$$

$$+ \int_{\partial\Omega} \begin{pmatrix} \boldsymbol{\sigma}(\overline{\boldsymbol{u}}^\dagger) \cdot \hat{\boldsymbol{n}} & \gamma_T(\overline{\boldsymbol{u}}^\dagger) \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}} \\ \gamma_T(\boldsymbol{u}) \end{pmatrix} d\partial\Omega. \tag{A.37}$$

A.2 Model space

A.2.1 Helmholtz I

The Gâteaux derivative of the Helmholtz I operator w.r.t. s^2, their adjoints and the corresponding boundary terms are respectively

$$\left\{\partial_{s^2} F\right\}(\delta s^2) = u\omega^2 \delta s^2, \quad \left\{\partial_{s^2}^\dagger F\right\}(u^\dagger) = \omega^2 \overline{u} u^\dagger, \quad \text{and} \quad \left[u^\dagger, \delta s^2\right]_{\partial_{s^2} F} = 0. \tag{A.38}$$

Indeed

$$\left\langle u^\dagger, \left\{\partial_{s^2} F\right\}(\delta s^2)\right\rangle_{L_2(\Omega)} = \left\langle u^\dagger, u\omega^2 \delta s^2\right\rangle_{L_2(\Omega)} = \left\langle \omega^2 \overline{u} u^\dagger, \delta s^2\right\rangle_{L_2(\Omega)}. \tag{A.39}$$

A.2.2 Helmholtz II

The Gâteaux derivative of the Helmholtz II operator w.r.t. ν and K, their adjoints and the corresponding boundary terms are respectively

$$\{\partial_\nu F\}(\delta\nu) = \operatorname{div}(\delta\nu\,\mathbf{grad}(p)), \tag{A.40}$$
$$\left\{\partial_\nu^\dagger F\right\}(p^\dagger) = -\mathbf{grad}(\overline{p})\cdot\mathbf{grad}\left(p^\dagger\right), \tag{A.41}$$
$$\left[p^\dagger, \delta\nu\right]_{\partial_\epsilon F} = \int_{\partial\Omega} \overline{p}^\dagger\,\delta\nu\frac{\partial p}{\partial n}\,d\partial\Omega \tag{A.42}$$

and

$$\{\partial_K F\}(\delta K) = p\omega^2\delta K, \tag{A.43}$$
$$\left\{\partial_K^\dagger F\right\}(p^\dagger) = \omega^2\overline{p}p^\dagger, \tag{A.44}$$
$$\left[p^\dagger, \delta K\right]_{\partial_K F} = 0. \tag{A.45}$$

Indeed

$$\left\langle p^\dagger, \{\partial_\nu F\}(\delta\nu)\right\rangle_{L_2(\Omega)} \tag{A.46}$$
$$= \left\langle p^\dagger, \operatorname{div}(\delta\nu\,\mathbf{grad}(p))\right\rangle \tag{A.47}$$
$$= \int_\Omega \overline{p}^\dagger \operatorname{div}(\delta\nu\,\mathbf{grad}(p))\;d\Omega \tag{A.48}$$
$$= -\int_\Omega \mathbf{grad}\left(\overline{p}^\dagger\right)\cdot\delta\nu\,\mathbf{grad}(p)\;d\Omega + \int_\Omega \operatorname{div}\left(\overline{p}^\dagger\delta\nu\,\mathbf{grad}(p)\right)\;d\Omega \tag{A.49}$$
$$= \left\langle -\mathbf{grad}\left(p^\dagger\right)\cdot\mathbf{grad}(\overline{p}), \delta\nu\right\rangle_{L_2(\Omega)} + \int_{\partial\Omega}\overline{p}^\dagger\,\delta\nu\frac{\partial p}{\partial n}\,d\partial\Omega \tag{A.50}$$

and

$$\left\langle p^\dagger, \{\partial_K F\}(\delta K)\right\rangle_{L_2(\Omega)} = \left\langle p^\dagger, p\omega^2\delta K\right\rangle_{L_2(\Omega)} = \left\langle \omega^2\overline{p}p^\dagger, \delta K\right\rangle_{L_2(\Omega)}. \tag{A.51}$$

A.2.3 Maxwell

The Gâteaux derivative of the Maxwell operator w.r.t. ϵ and μ and their adjoints are respectively

$$\{\partial_\epsilon F\}(\delta\epsilon) = -i\boldsymbol{e}\omega\delta\epsilon, \quad \left\{\partial_\epsilon^\dagger F\right\}(u^\dagger) = i\omega\overline{\boldsymbol{e}}\cdot\boldsymbol{e}^\dagger, \quad \left[u^\dagger, \delta\epsilon\right]_{\partial_\epsilon F} = 0 \tag{A.52}$$

and

$$\{\partial_\mu F\}(\delta\mu) = i\boldsymbol{h}\omega\delta\mu, \quad \left\{\partial_\mu^\dagger F\right\}(u^\dagger) = -i\omega\overline{\boldsymbol{h}}\cdot\boldsymbol{h}^\dagger, \quad \left[u^\dagger, \delta\mu\right]_{\partial_\mu F} = 0. \tag{A.53}$$

Indeed

$$\left\langle u^\dagger, \{\partial_\epsilon F\}(\delta\epsilon)\right\rangle_{L_2^3(\Omega)\times L_2^3(\Omega)} = \left\langle \boldsymbol{e}^\dagger, -i\boldsymbol{e}\omega\delta\epsilon\right\rangle_{L_2^3(\Omega)} = \left\langle i\omega\boldsymbol{e}^\dagger\cdot\overline{\boldsymbol{e}}, \delta\epsilon\right\rangle_{L_2(\Omega)} \tag{A.54}$$

and

$$\left\langle u^\dagger, \{\partial_\mu F\}(\delta\mu)\right\rangle_{L_2^3(\Omega)\times L_2^3(\Omega)} = \left\langle \boldsymbol{h}^\dagger, i\omega\boldsymbol{h}\delta\mu\right\rangle_{L_2^3(\Omega)} = \left\langle -i\omega\boldsymbol{h}^\dagger\cdot\overline{\boldsymbol{h}}, \delta\mu\right\rangle_{L_2(\Omega)}. \tag{A.55}$$

A.2.4 Navier

The Gâteaux derivative of the Navier operator w.r.t. ρ, λ and μ, their adjoints and the corresponding boundary terms are

$$\{\partial_\rho F\}(\delta\rho) = \boldsymbol{u}\omega^2\delta\rho, \tag{A.56}$$
$$\left\{\partial_\rho^\dagger F\right\}(\boldsymbol{u}^\dagger) = \omega^2\overline{\boldsymbol{u}}\cdot\boldsymbol{u}^\dagger, \tag{A.57}$$
$$\left[u^\dagger, \delta\rho\right]_{\partial_\rho F} = 0 \tag{A.58}$$

and

$$\{\partial_\lambda F\}(\delta\lambda) = \operatorname{div}\left(\{\partial_\lambda\boldsymbol{\sigma}\}(\delta\lambda)\right), \tag{A.59}$$
$$\left\{\partial_\lambda^\dagger F\right\}(\boldsymbol{u}^\dagger) = -\operatorname{div}(\overline{\boldsymbol{u}})\operatorname{div}\left(\boldsymbol{u}^\dagger\right), \tag{A.60}$$
$$\left[u^\dagger, \delta\lambda\right]_{\partial_\lambda F} = \int \hat{\boldsymbol{n}}\cdot\{\partial_\lambda\boldsymbol{\sigma}\}(\delta\lambda)\cdot\overline{\boldsymbol{u}}^\dagger\, d\partial\Omega \tag{A.61}$$

and

$$\{\partial_\mu F\}(\delta\mu) = \operatorname{div}\left(\{\partial_\mu\boldsymbol{\sigma}\}(\delta\mu)\right), \tag{A.62}$$
$$\left\{\partial_\mu^\dagger F\right\}(\boldsymbol{u}^\dagger) = -2\boldsymbol{\epsilon}(\overline{\boldsymbol{u}}) : \boldsymbol{\epsilon}(\boldsymbol{u}^\dagger), \tag{A.63}$$
$$\left[u^\dagger, \delta\mu\right]_{\partial_\mu F} = \int \hat{\boldsymbol{n}}\cdot\{\partial_\mu\boldsymbol{\sigma}\}(\delta\mu)\cdot\overline{\boldsymbol{u}}^\dagger\, d\partial\Omega. \tag{A.64}$$

Indeed

$$\left\langle u^\dagger, \{\partial_\rho F\}(\delta\rho)\right\rangle_{L_2^3(\Omega)} = \left\langle \boldsymbol{u}^\dagger, \omega^2\delta\rho\boldsymbol{u}\right\rangle_{L_2^3(\Omega)} = \left\langle \omega^2\boldsymbol{u}^\dagger\cdot\overline{\boldsymbol{u}}, \delta\rho\right\rangle_{L_2(\Omega)} \tag{A.65}$$
$$\left\langle u^\dagger, \{\partial_\lambda F\}(\delta\lambda)\right\rangle_{L_2^3(\Omega)} = \left\langle \boldsymbol{u}^\dagger, \operatorname{div}\left(\{\partial_\lambda\boldsymbol{\sigma}\}(\delta\lambda)\right)\right\rangle \tag{A.66}$$
$$= \int_\Omega \overline{\boldsymbol{u}}^\dagger\cdot\operatorname{div}\left(\{\partial_\lambda\boldsymbol{\sigma}\}(\delta\lambda)\right)\, d\Omega \tag{A.67}$$
$$= \int_\Omega \operatorname{div}\left(\{\partial_\lambda\boldsymbol{\sigma}\}(\delta\lambda)\cdot\overline{\boldsymbol{u}}^\dagger\right)\, d\Omega - \int_\Omega \mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) : \{\partial_\lambda\boldsymbol{\sigma}\}(\delta\lambda)\, d\Omega \tag{A.68}$$

$$= -\int_\Omega \operatorname{div}\left(\overline{\boldsymbol{u}}^\dagger\right) \operatorname{div}(\boldsymbol{u})\, \delta\lambda \, d\Omega + \int_{\partial\Omega} \hat{\boldsymbol{n}} \cdot \{\partial_\lambda \boldsymbol{\sigma}\}(\delta\lambda) \cdot \overline{\boldsymbol{u}}^\dagger \, d\partial\Omega \tag{A.69}$$

$$= \left\langle -\operatorname{div}(\overline{\boldsymbol{u}}) \operatorname{div}\left(\boldsymbol{u}^\dagger\right), \delta\lambda \right\rangle_{L_2(\Omega)} + \int_{\partial\Omega} \hat{\boldsymbol{n}} \cdot \{\partial_\lambda \boldsymbol{\sigma}\}(\delta\lambda) \cdot \overline{\boldsymbol{u}}^\dagger \, d\partial\Omega \tag{A.70}$$

and similarly

$$\left\langle u^\dagger, \{\partial_\mu F\}(\delta\mu) \right\rangle_{L_2^3(\Omega)} = \left\langle \boldsymbol{u}^\dagger, \operatorname{div}\left(\{\partial_\mu \boldsymbol{\sigma}\}(\delta\mu)\right) \right\rangle \tag{A.71}$$

$$= \int_\Omega \overline{\boldsymbol{u}}^\dagger \cdot \operatorname{div}\left(\{\partial_\mu \boldsymbol{\sigma}\}(\delta\mu)\right) \, d\Omega \tag{A.72}$$

$$= \int_\Omega \operatorname{div}\left(\{\partial_\mu \boldsymbol{\sigma}\}(\delta\mu) \cdot \overline{\boldsymbol{u}}^\dagger\right) \, d\Omega - \int_\Omega \mathbf{grad}\left(\overline{\boldsymbol{u}}^\dagger\right) : \{\partial_\mu \boldsymbol{\sigma}\}(\delta\mu) \, d\Omega \tag{A.73}$$

$$= -\int_\Omega 2\boldsymbol{\epsilon}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\overline{\boldsymbol{u}}^\dagger)\delta\mu \, d\Omega + \int_{\partial\Omega} \hat{\boldsymbol{n}} \cdot \{\partial_\mu \boldsymbol{\sigma}\}(\delta\mu) \cdot \overline{\boldsymbol{u}}^\dagger \, d\partial\Omega \tag{A.74}$$

$$= \left\langle -2\boldsymbol{\epsilon}(\overline{\boldsymbol{u}}) : \boldsymbol{\epsilon}(\boldsymbol{u}^\dagger), \delta\mu \right\rangle_{L_2(\Omega)} + \int_{\partial\Omega} \hat{\boldsymbol{n}} \cdot \{\partial_\mu \boldsymbol{\sigma}\}(\delta\mu) \cdot \overline{\boldsymbol{u}}^\dagger \, d\partial\Omega. \tag{A.75}$$

A.3 Data space

A.3.1 Gradient kernel of the least squared distance

Consider the least squares distance

$$H(u) = \frac{1}{2} \sum_r |u(\boldsymbol{x}_r) - d_r|^2 . \tag{A.76}$$

Its Gâteaux directional derivative writes

$$\{D_u H(u)\}(\delta u) = \frac{1}{2} \sum_r \overline{(u(\boldsymbol{x}_r) - d_r)}\, \delta u(\boldsymbol{x}_r) + (u(\boldsymbol{x}_r) - d_r)\, \overline{\delta u(\boldsymbol{x}_r)} \tag{A.77}$$

$$= \operatorname{Re} \sum_r \overline{(u(\boldsymbol{x}_r) - d_r)}\, \delta u(\boldsymbol{x}_r). \tag{A.78}$$

Then using the identity

$$\delta u(\boldsymbol{x}_r) = \int_\Omega \delta u(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{x}_r) \, d\boldsymbol{x}, \tag{A.79}$$

the gradient kernel is found as

$$\{D_u H(u)\}(\delta u) = \operatorname{Re} \sum_r \overline{(u(\boldsymbol{x}_r) - d_r)} \delta u(\boldsymbol{x}_r) \tag{A.80}$$

$$= \operatorname{Re} \sum_r \overline{(u(\boldsymbol{x}_r) - d_r)} \int_\Omega \delta u(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{x}_r) \, d\boldsymbol{x} \tag{A.81}$$

$$= \mathrm{Re} \int_\Omega \sum_r \overline{(u(\boldsymbol{x}_r) - d_r)} \delta(\boldsymbol{x} - \boldsymbol{x}_r)\, \delta u(\boldsymbol{x})\, d\boldsymbol{x} \tag{A.82}$$

$$= \mathrm{Re} \left\langle \sum_r (u(\boldsymbol{x}_r) - d_r) \delta(\boldsymbol{x} - \boldsymbol{x}_r), \delta u \right\rangle \tag{A.83}$$

$$:= \mathrm{Re} \left\langle h', \delta u \right\rangle . \tag{A.84}$$

Appendix B

Extended numerical illustration

This appendix aims to reproduce the results from Section 1.6 in Chapter 1 for the Helmholtz II equation, the Maxwell's equations and the Navier equation. For conciseness, only the highly penetrable case, where the cylinder is modeled explicitly, is considered and only a single scale is used for the Sobolev gradient. For all the numerical illustrations, the pulsation ω is again set to 2π.

B.1 Helmholtz II

The systematic steps of the adjoint state method are as follows, in terms of two bulk model parameters ν and K and a boundary model parameter β:

1. Find the direct states p_e obeying the direct problem

$$\begin{cases} \operatorname{div}\left(\nu\,\mathbf{grad}\left(p_e\right)\right) + \omega^2 K p_e &= \delta(\boldsymbol{x}-\boldsymbol{x}_e) \quad \text{in } \Omega \\ \nu\dfrac{\partial p_e}{\partial n} + \beta p_e &= 0 \qquad\qquad\quad \text{on } \partial\Omega. \end{cases} \tag{B.1}$$

2. Find the adjoint states $p_e^\dagger$ obeying the adjoint problem

$$\begin{cases} \operatorname{div}\left(\nu\,\mathbf{grad}\left(\overline{p}_e^\dagger\right)\right) + \omega^2 K \overline{p}_e^\dagger &= \displaystyle\sum_r (\overline{p}_e(\boldsymbol{x}_r) - \overline{d}_{e,r})\delta(\boldsymbol{x}-\boldsymbol{x}_r) \quad \text{in } \Omega \\ \nu\dfrac{\partial \overline{p}_e^\dagger}{\partial n} + \beta \overline{p}_e^\dagger &= 0 \qquad\qquad\qquad\qquad\quad \text{on } \partial\Omega. \end{cases} \tag{B.2}$$

3. Once the direct and adjoint states are known, the gradient kernels with respect to the different model parameters can be evaluated:

   - bulk sensitivity w.r.t. ν

$$\vec{j}'_\Omega = \sum_e \mathbf{grad}\left(\overline{p}_e^\dagger\right) \cdot \mathbf{grad}\left(p_e\right) \tag{B.3}$$

- bulk sensitivity w.r.t. K

$$\bar{j}'_{\Omega} = -\omega^2 \sum_e \overline{p}_e^{\dagger} p_e \tag{B.4}$$

- boundary sensitivity w.r.t. β

$$\bar{j}'_{\partial\Omega} = \sum_e \overline{p}_e^{\dagger} p_e. \tag{B.5}$$

4. Find the perturbed direct states δp_e obeying the perturbed direct problem

$$\begin{cases} \operatorname{div}(\nu \, \mathbf{grad}(\delta p_e)) + \omega^2 K \delta p_e & = -\operatorname{div}(\delta\nu \, \mathbf{grad}(p_e)) - \omega^2 \delta K p_e \quad \text{in } \Omega \\ \nu \dfrac{\partial \delta p_e}{\partial n} + \beta \delta p_e & = -\delta\nu \dfrac{\partial p_e}{\partial n} - \delta\beta p_e \qquad\qquad \text{on } \partial\Omega. \end{cases} \tag{B.6}$$

5. Find the perturbed adjoint states $\delta u_e^{\dagger}$ obeying the perturbed adjoint problem

$$\begin{cases} \operatorname{div}\left(\nu \, \mathbf{grad}\left(\delta \overline{p}_e^{\dagger}\right)\right) + \omega^2 K \delta \overline{p}_e^{\dagger} & = \displaystyle\sum_r \delta \overline{p}_e(\boldsymbol{x}_r)\delta(\boldsymbol{x} - \boldsymbol{x}_r) \\ & \quad - \operatorname{div}\left(\delta\nu \, \mathbf{grad}\left(\overline{p}_e^{\dagger}\right)\right) - \omega^2 \delta K \overline{p}_e^{\dagger} \quad \text{in } \Omega \\ \nu \dfrac{\partial \delta \overline{p}_e^{\dagger}}{\partial n} + \beta \delta \overline{p}_e^{\dagger} & = -\delta\nu \dfrac{\partial \overline{p}_e^{\dagger}}{\partial n} - \delta\beta \overline{p}_e^{\dagger} \qquad\qquad \text{on } \partial\Omega. \end{cases} \tag{B.7}$$

6. Once the perturbed direct and adjoint states are known, the application of the Hessian operator in a particular direction can be evaluated:

- perturbed bulk sensitivity in the direction $\delta\nu$

$$\delta\bar{j}'_{\Omega} = \sum_e \mathbf{grad}\left(\delta \overline{p}_e^{\dagger}\right) \cdot \mathbf{grad}(p_e) + \mathbf{grad}\left(\overline{p}_e^{\dagger}\right) \cdot \mathbf{grad}(\delta p_e) \tag{B.8}$$

- perturbed bulk sensitivity in the direction δK

$$\delta\bar{j}'_{\Omega} = -\omega^2 \sum_e \delta \overline{p}_e^{\dagger} p_e + \overline{p}_e^{\dagger} \delta p_e \tag{B.9}$$

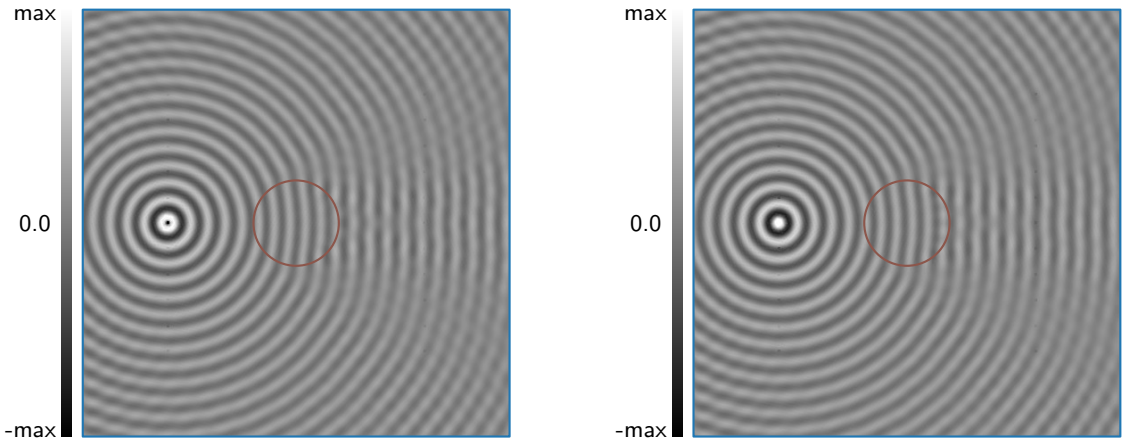
- perturbed boundary sensitivity in the direction $\delta\beta$

$$\delta\bar{j}'_{\partial\Omega} = \sum_e \delta \overline{p}_e^{\dagger} p_e + \overline{p}_e^{\dagger} \delta p_e. \tag{B.10}$$

The unbounded propagation domain is again modeled by means of an absorbing boundary condition

$$\nu \frac{\partial p_e}{\partial n} + \beta_0(\nu, K) p_e = 0. \tag{B.11}$$

For simplicity, a zeroth order absorbing condition has been chosen, for which the relationship $\beta_0(\nu, K) := i\omega\sqrt{\nu K}$ holds. Derivatives w.r.t. ν or K can easily be obtained from the derivatives w.r.t. β through the chain rule.



(a) $\mathrm{Re}\,(p)$ (b) $\mathrm{Im}\,(p)$

Figure B.1: Ground-truth direct field p for a highly penetrable cylinder and for a single source. The background and cylinder specific volume and compressibility are respectively $\tilde{\nu}_c = 0.9$, $\tilde{K}_c = 1.1$ and $\tilde{\nu}_0 = 1.0$, $\tilde{K}_0 = 1.0$.

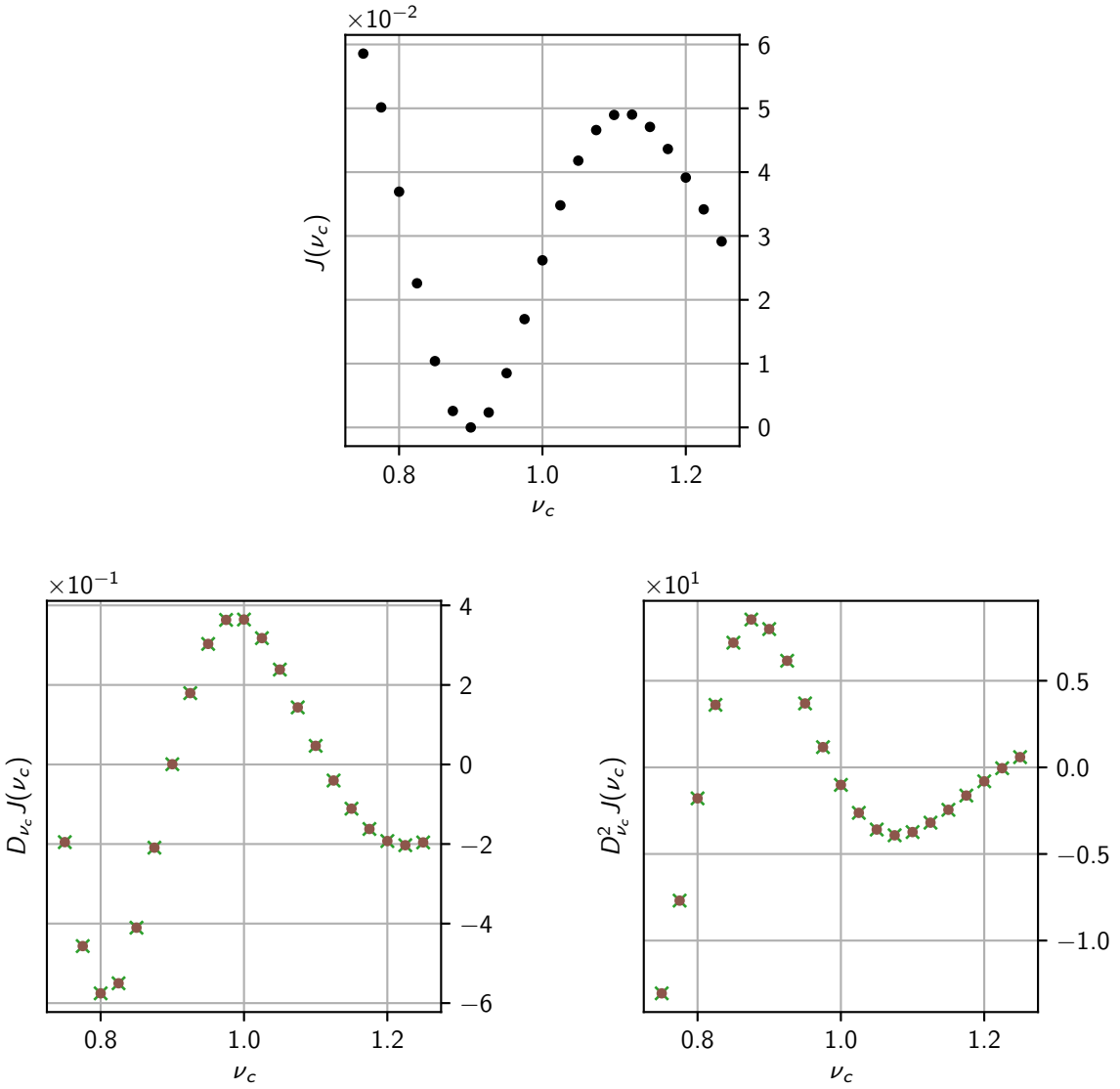


Figure B.2: Performance functional (•) and its derivatives for a highly penetrable cylinder whose specific volume ν_c varies around the ground-truth value $\tilde{\nu}_c = 0.9$ while the compressibility is kept at the ground-truth value $\tilde{K}_c = 1.1$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

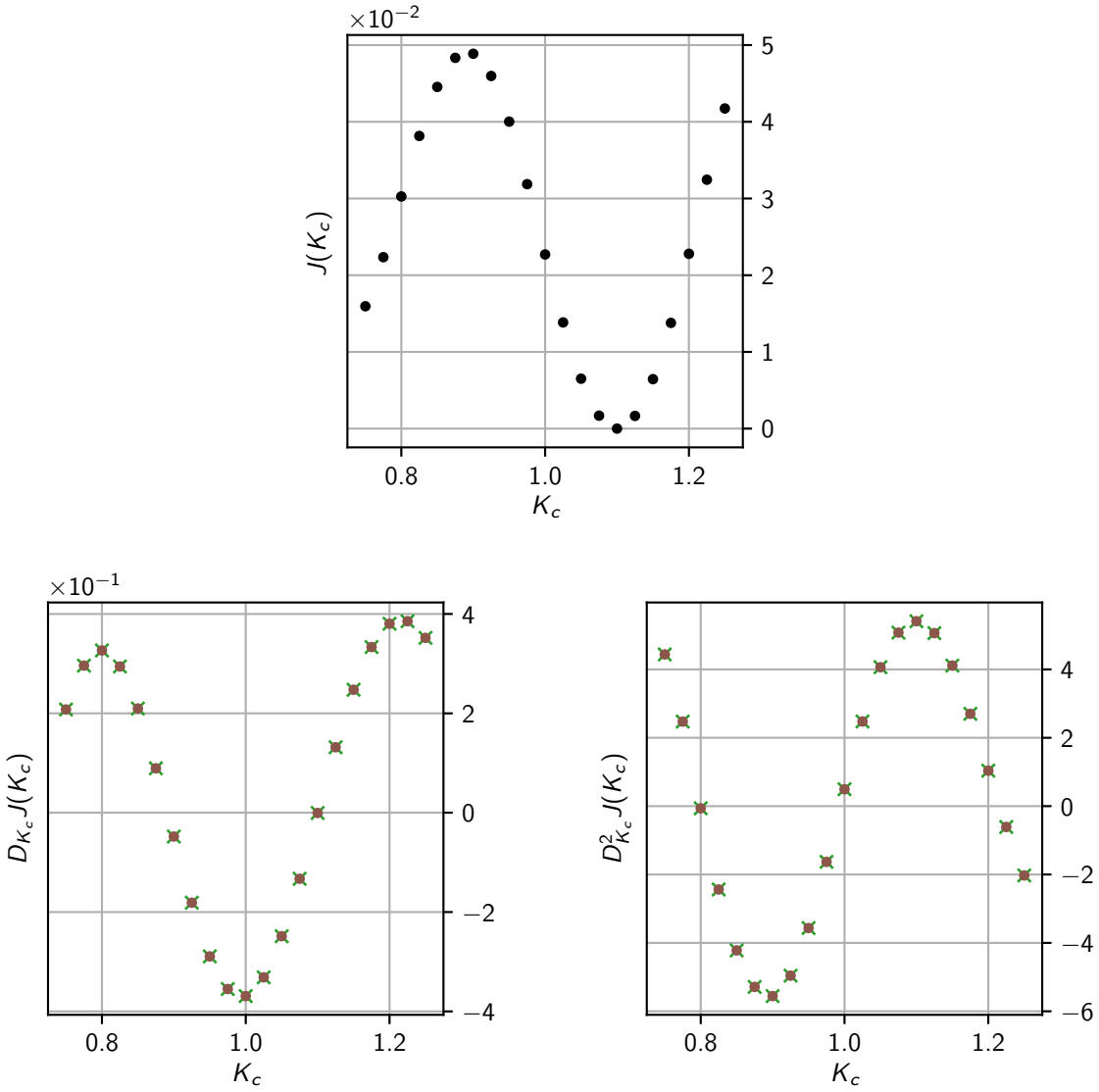


Figure B.3: Performance functional (•) and its derivatives for a highly penetrable cylinder whose compressibility K_c varies around the ground-truth value $\tilde{K}_c = 1.1$ while the specific volume is kept at the ground-truth value $\tilde{\nu}_c = 0.9$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

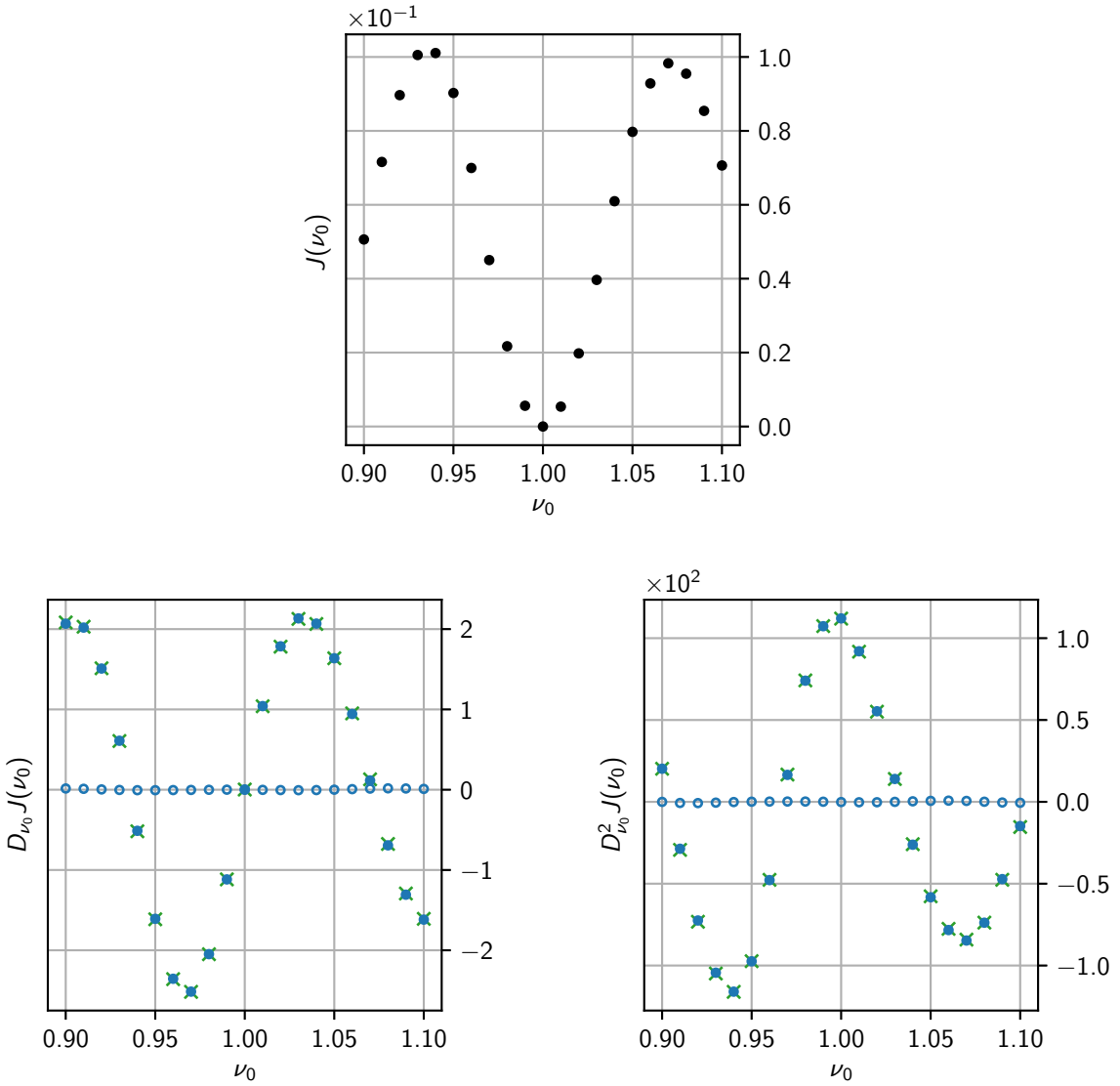


Figure B.4: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\nu}_c = 0.9$, $\tilde{K}_c = 1.1$) embedded in a background medium whose specific volume ν_0 varies around the ground-truth value $\tilde{\nu}_0 = 1.0$ while the compressibility is kept at the ground-truth value $\tilde{K}_0 = 1.0$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

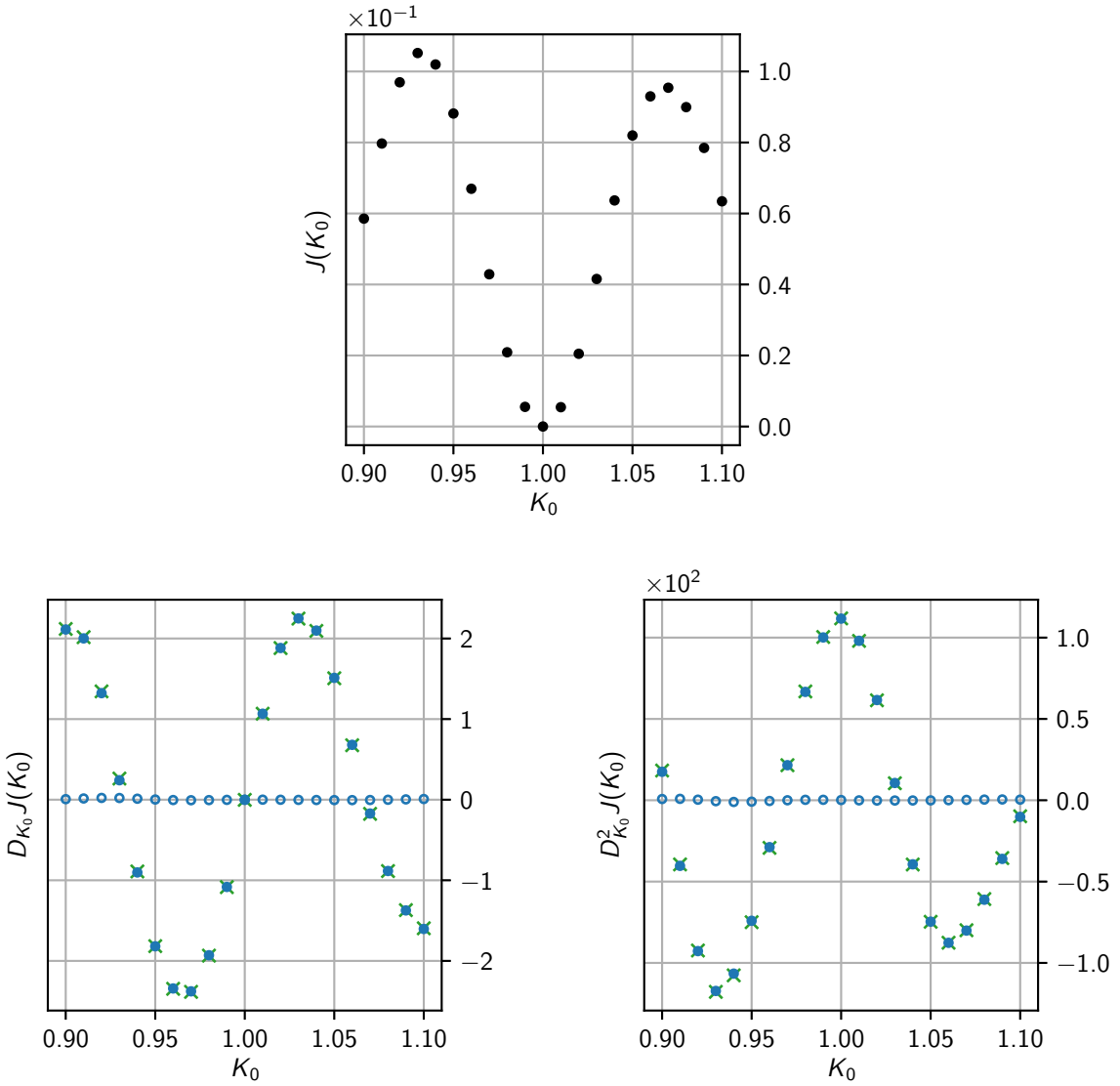


Figure B.5: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\nu}_c = 0.9$, $\tilde{K}_c = 1.1$) embedded in a background medium whose compressibility K_0 varies around the ground-truth value $\tilde{K}_0 = 1.0$ while the specific volume is kept at the ground-truth value $\tilde{\nu}_0 = 1.0$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

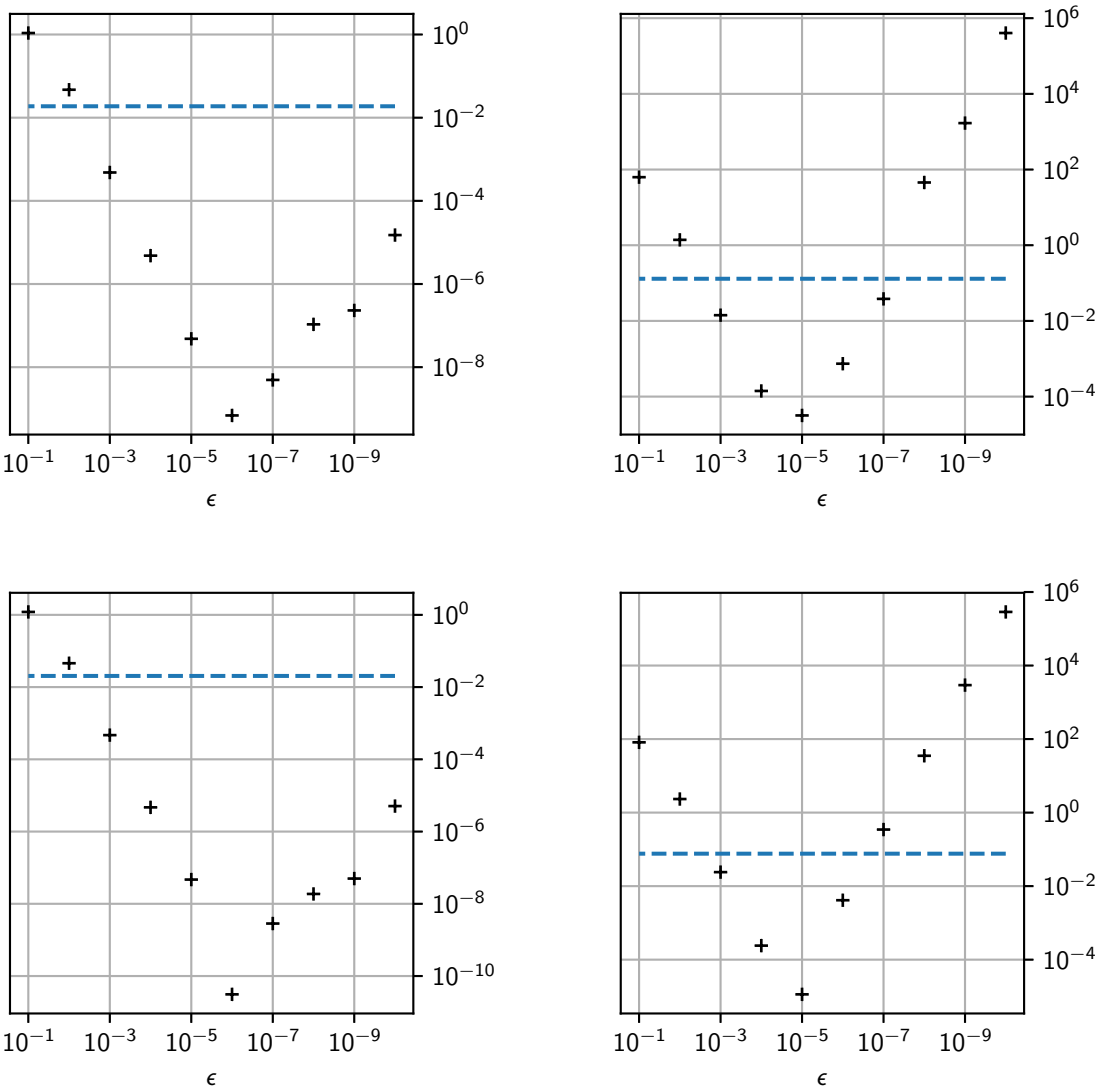


Figure B.6: Difference between the approximation of (1.3) and the adjoint state method with the boundary contribution for a highly penetrable cylinder ($\tilde{\nu}_c = 0.9$, $\tilde{K}_c = 1.1$) when the unknown background specific volume and compressibility are respectively $\nu_0 = 1.04$ and $K_0 = 0.96$ while the ground-truth value are respectively $\tilde{\nu}_0 = 1.0$ and $\tilde{K}_0 = 1.0$. Top row is related to specific volume derivatives, bottom row to compressibility derivatives. Left column is related to first order derivatives, right column to second order derivatives. The dashed blue line (——) is the amplitude of the boundary contribution in the adjoint state method.

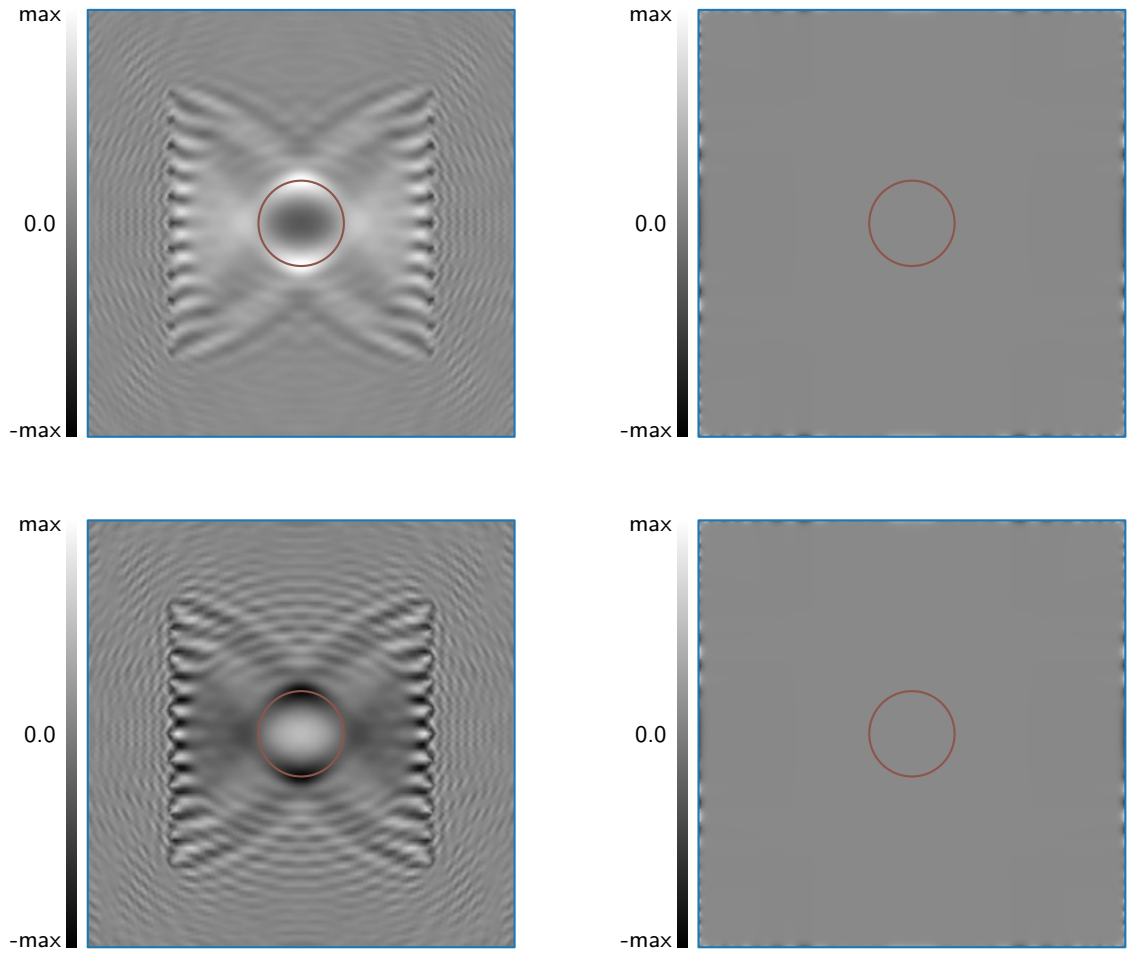


Figure B.7: Bulk (left column) and boundary (right column) parts of the H_1-Sobolev gradient ($l_c := 2\pi\sqrt{\alpha_1}$; $l_c = 1$). The data used for this gradient is obtained with a highly penetrable cylinder ($\tilde{\nu}_c = 0.9$, $\tilde{\nu}_0 = 1.0$ and $\tilde{K}_c = 1.1$, $\tilde{K}_0 = 1.0$). Top row is specific volume gradients. Bottom row is compressibility gradients. From top to bottom, the maximal value for the bulk and boundary contributions are respectively 6×10^{-5}, 2.3×10^{-5} (bulk) and 1.5×10^{-5}, 1.8×10^{-6} (boundary).

B.2 Maxwell

Maxwell's equations are unchanged when $\boldsymbol{e}$ and ϵ are substituted by $\boldsymbol{h}$ and $-\mu$ respectively. For that reason, only permittivity perturbations and electric field measurements are considered as the derivations for permeability perturbations and magnetic field measurements are very similar. However, a conductivity parameter σ is added $\epsilon \to \epsilon - i\frac{\sigma}{\omega}$. Mathematically, all the derivatives w.r.t. σ can be obtained by the chain rule but the formulas are nevertheless given here explicitly because they are often used in practice. The absorbing boundary condition is also formulated from the point of view of the electric field $\boldsymbol{e}$, *i.e.*

$$\gamma_t(\boldsymbol{h}) + \beta\gamma_T(\boldsymbol{e}) = 0. \tag{B.12}$$

Note that because only permittivity pertubations and electric field measurements are considered, the problem could be expressed without the magnetic field appearing explicitly, *i.e.* as a single second order equation instead of two first order equations.
The systematic steps of the adjoint state method are as follows, in terms of two bulk model parameters ϵ and σ and a boundary model parameter β:

1. Find the direct states $(\boldsymbol{e}_e, \boldsymbol{h}_e)$ obeying the direct problem

$$\begin{cases} \mathbf{curl}\,(\boldsymbol{h}_e) - i\omega\epsilon\boldsymbol{e}_e & = \hat{\boldsymbol{f}}_{\boldsymbol{e}}\delta(\boldsymbol{x}-\boldsymbol{x}_e) & \text{in } \Omega \\ \mathbf{curl}\,(\boldsymbol{e}_e) + i\omega\mu\boldsymbol{h}_e & = \hat{\boldsymbol{f}}_{\boldsymbol{h}}\delta(\boldsymbol{x}-\boldsymbol{x}_e) & \text{in } \Omega \\ \gamma_t(\boldsymbol{h}_e) + \beta\gamma_T(\boldsymbol{e}_e) & = 0 & \text{on } \partial\Omega. \end{cases} \tag{B.13}$$

2. Find the adjoint states $\left(\overline{\boldsymbol{e}}_e^\dagger, \overline{\boldsymbol{h}}_e^\dagger\right)$ obeying the adjoint problem

$$\begin{cases} \mathbf{curl}\left(\overline{\boldsymbol{h}}_e^\dagger\right) - i\omega\epsilon\overline{\boldsymbol{e}}_e^\dagger & = \sum_r (\overline{\boldsymbol{e}}_e(\boldsymbol{x}_r) - \overline{d}_{e,r})\delta(\boldsymbol{x}-\boldsymbol{x}_r) & \text{in } \Omega \\ \mathbf{curl}\left(\overline{\boldsymbol{e}}_e^\dagger\right) + i\omega\mu\overline{\boldsymbol{h}}_e^\dagger & = 0 & \text{in } \Omega \\ \gamma_t(\overline{\boldsymbol{h}}_e^\dagger) + \beta\gamma_T(\overline{\boldsymbol{e}}_e^\dagger) & = 0 & \text{on } \partial\Omega. \end{cases} \tag{B.14}$$

3. Once the direct and adjoint states are known, the gradient kernels with respect to the different model parameters can be evaluated:

   - bulk sensitivity w.r.t. ϵ

$$\vec{J}'_\Omega = -i\omega \sum_e \overline{\boldsymbol{e}}_e^\dagger \cdot \boldsymbol{e}_e \tag{B.15}$$

   - bulk sensitivity w.r.t. σ

$$\vec{J}'_\Omega = -\sum_e \overline{\boldsymbol{e}}_e^\dagger \cdot \boldsymbol{e}_e \tag{B.16}$$

- boundary sensitivity w.r.t. β

$$\bar{j}'_{\partial\Omega} = \sum_e \gamma_T(\overline{\boldsymbol{e}}_e^\dagger) \cdot \gamma_T(\boldsymbol{e}_e). \tag{B.17}$$

4. Find the perturbed direct states $(\delta\boldsymbol{e}_e, \delta\boldsymbol{h}_e)$ obeying the perturbed direct problem

$$\begin{cases} \mathbf{curl}\,(\delta\boldsymbol{h}_e) - i\omega\epsilon\delta\boldsymbol{e}_e &= i\omega\delta\epsilon\boldsymbol{e}_e & \text{in } \Omega \\ \mathbf{curl}\,(\delta\boldsymbol{e}_e) + i\omega\mu\delta\boldsymbol{h}_e &= 0 & \text{in } \Omega \\ \gamma_t(\delta\boldsymbol{h}_e) + \beta\gamma_T(\delta\boldsymbol{e}_e) &= -\delta\beta\gamma_T(\boldsymbol{e}_e) & \text{on } \partial\Omega. \end{cases} \tag{B.18}$$

5. Find the perturbed adjoint states $\left(\delta\overline{\boldsymbol{e}}_e^\dagger, \delta\overline{\boldsymbol{h}}_e^\dagger\right)$ obeying the adjoint problem

$$\begin{cases} \mathbf{curl}\left(\delta\overline{\boldsymbol{h}}_e^\dagger\right) - i\omega\epsilon\delta\overline{\boldsymbol{e}}_e^\dagger &= \sum_r \delta\overline{\boldsymbol{e}}_e(\boldsymbol{x}_r)\delta(\boldsymbol{x}-\boldsymbol{x}_r) + i\omega\delta\epsilon\overline{\boldsymbol{e}}_e^\dagger & \text{in } \Omega \\ \mathbf{curl}\left(\delta\overline{\boldsymbol{e}}_e^\dagger\right) + i\omega\mu\delta\overline{\boldsymbol{h}}_e^\dagger &= 0 & \text{in } \Omega \\ \gamma_t(\delta\overline{\boldsymbol{h}}_e^\dagger) + \beta\gamma_T(\delta\overline{\boldsymbol{e}}_e^\dagger) &= -\delta\beta\gamma_T(\overline{\boldsymbol{e}}_e^\dagger) & \text{on } \partial\Omega. \end{cases} \tag{B.19}$$

6. Once the perturbed direct and adjoint states are known, the application of the Hessian operator in a particular direction can be evaluated:

- second order bulk sensitivity w.r.t. ϵ

$$\delta\bar{j}'_\Omega = -i\omega \sum_e \delta\overline{\boldsymbol{e}}_e^\dagger \cdot \boldsymbol{e}_e + \overline{\boldsymbol{e}}_e^\dagger \cdot \delta\boldsymbol{e}_e \tag{B.20}$$

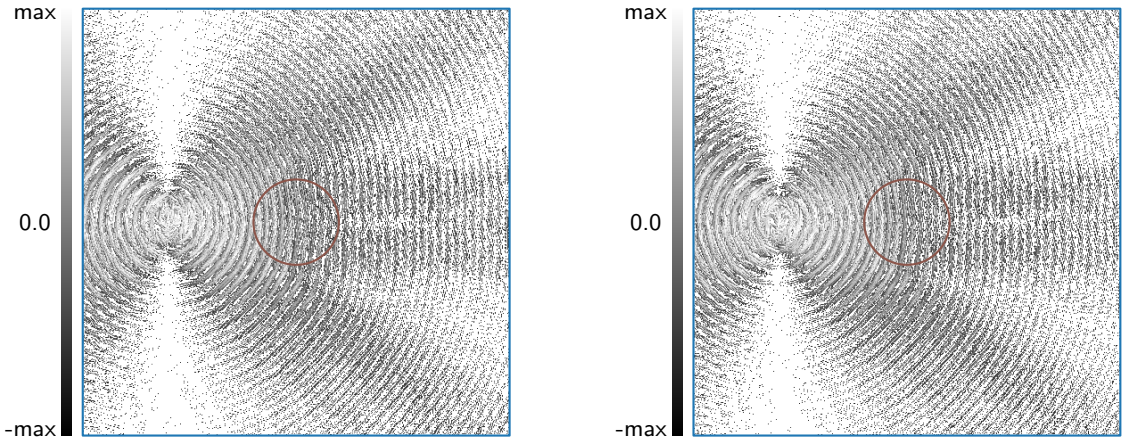
- second order bulk sensitivity w.r.t. σ

$$\delta\bar{j}'_\Omega = -\sum_e \delta\overline{\boldsymbol{e}}_e^\dagger \cdot \boldsymbol{e}_e + \overline{\boldsymbol{e}}_e^\dagger \cdot \delta\boldsymbol{e}_e \tag{B.21}$$

- second order boundary sensitivity w.r.t. β

$$\delta\bar{j}'_{\partial\Omega} = \sum_e \gamma_T(\delta\overline{\boldsymbol{e}}_e^\dagger) \cdot \gamma_T(\boldsymbol{e}_e) + \gamma_T(\overline{\boldsymbol{e}}_e^\dagger) \cdot \gamma_T(\delta\boldsymbol{e}_e). \tag{B.22}$$

For simplicity, the unbounded propagation domain is again modeled by means of a zeroth order absorbing condition, for which the relationship $\beta_0(\epsilon, \sigma, \mu) := -\sqrt{\frac{\epsilon - i\sigma/\omega}{\mu}}$ holds. Derivatives w.r.t. ϵ and σ can easily be obtained from the derivatives w.r.t. β through the chain rule. For the numerical illustrations, the magnetic field is not excited, *i.e.* $\hat{\boldsymbol{f}}_{\boldsymbol{h}} = 0$. Electric field excitation is polarized along the y-axis, *i.e.* $\hat{\boldsymbol{f}}_{\boldsymbol{e}} = \hat{\boldsymbol{y}}$, and only the y-component of the electric field is measured.



(a) $\mathrm{Re}(\boldsymbol{e})$ (b) $\mathrm{Im}(\boldsymbol{e})$

Figure B.8: Ground-truth direct field $\boldsymbol{e}$ for a highly penetrable cylinder and for a single source. The background and cylinder permittivity and conductivity are respectively $\tilde{\epsilon}_c = 1.$, $\tilde{\sigma}_c = \pi \times 10^{-1}$ and $\tilde{\epsilon}_0 = 1.2$, $\tilde{\sigma}_0 = 2\pi \times 10^{-2}$.

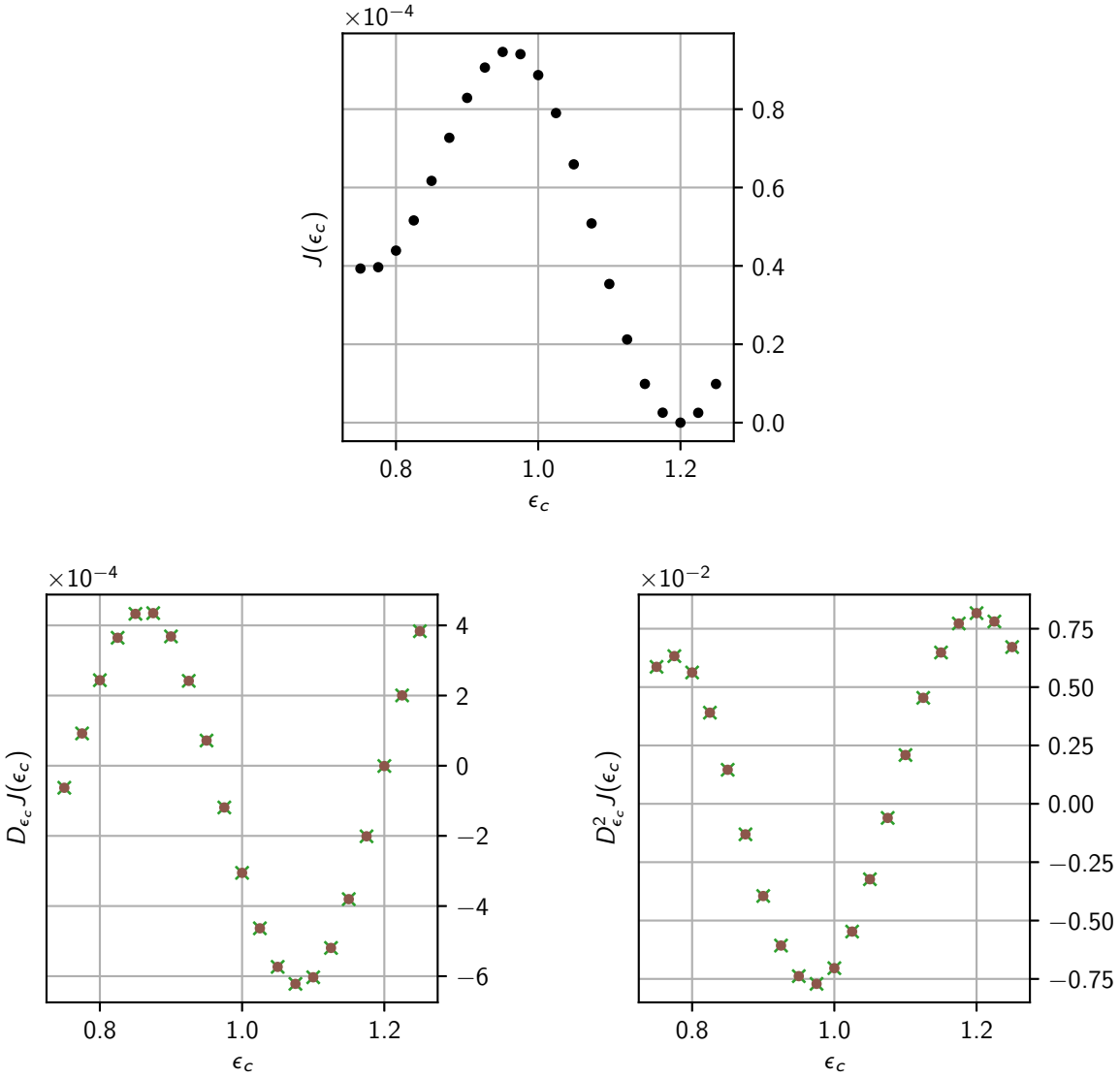


Figure B.9: Performance functional (•) and its derivatives for a highly penetrable cylinder, embedded in a background medium ($\tilde{\epsilon}_0 = 1.2$, $\tilde{\sigma}_0 = 2\pi \times 10^{-2}$), whose permittivity ϵ_c varies around the ground-truth value $\tilde{\epsilon}_c = 1.2$ while the conductivity is kept at the ground-truth value $\tilde{\sigma}_c = 2\pi \times 10^{-2}$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

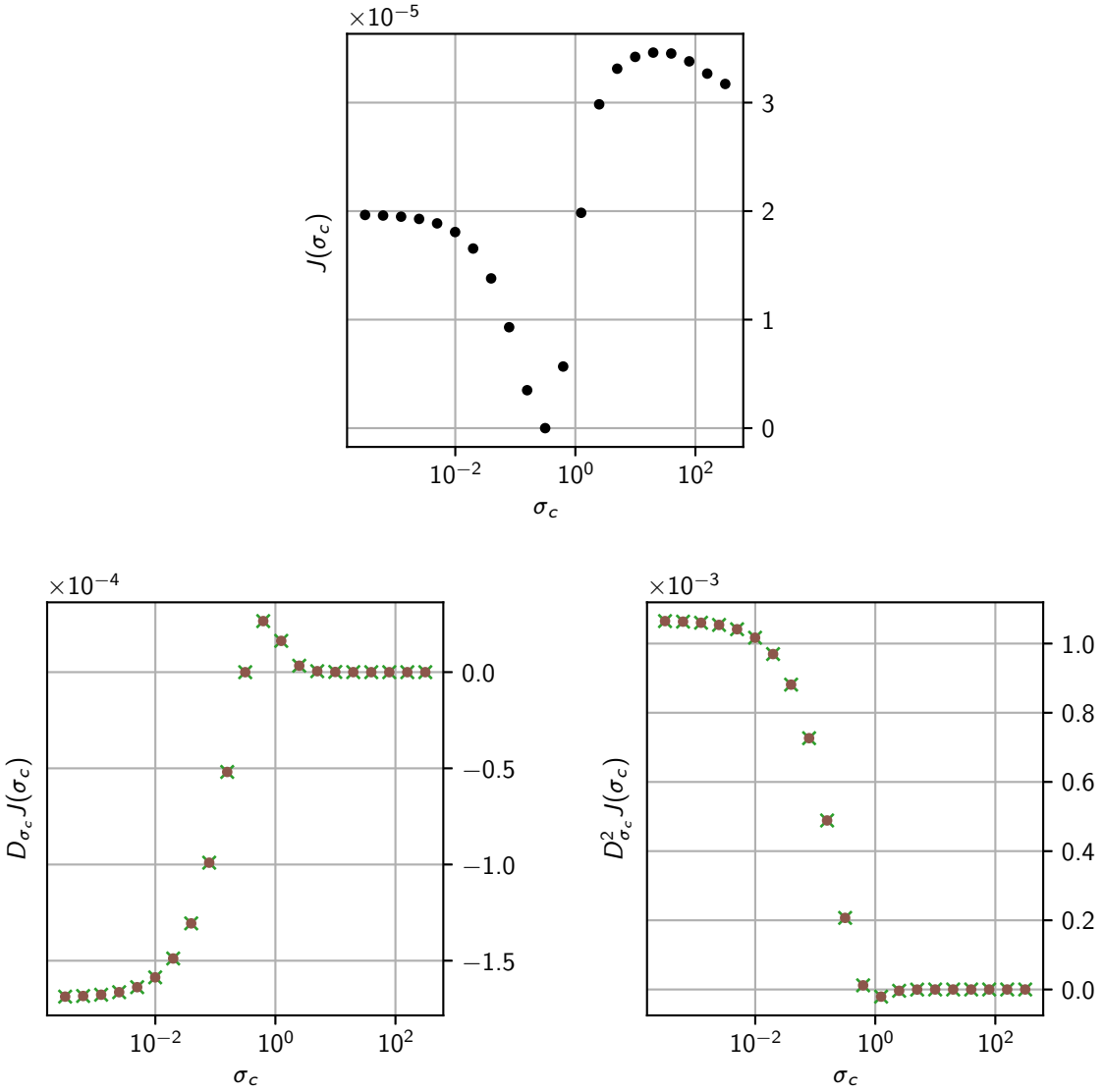


Figure B.10: Performance functional (•) and its derivatives for a highly penetrable cylinder, embedded in a background medium ($\tilde{\epsilon}_0 = 1.2$, $\tilde{\sigma}_0 = 2\pi \times 10^{-2}$), whose conductivity σ_c varies around the ground-truth value $\tilde{\sigma}_c = \pi \times 10^{-1}$ while the permittivity is kept at the ground-truth value $\tilde{\epsilon}_c = 1.2$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

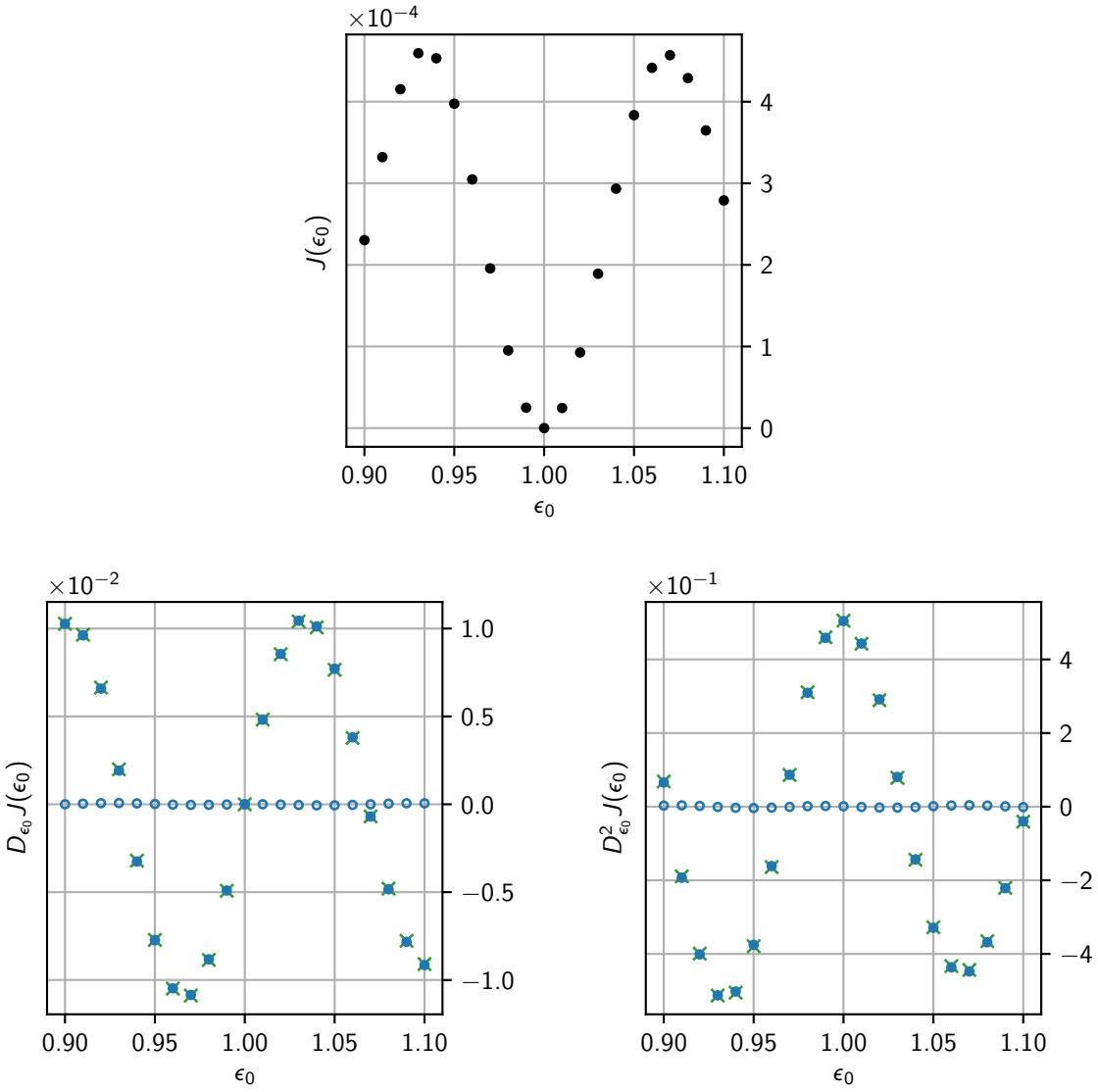


Figure B.11: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\epsilon}_c = 1.2$, $\tilde{\epsilon}_c = \pi \times 10^{-1}$) embedded in a background medium whose permittivity ϵ_0 varies around the ground-truth value $\tilde{\epsilon}_0 = 1.2$ while the conductivity is kept at the ground-truth value $\tilde{\sigma}_0 = 2\pi \times 10^{-2}$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

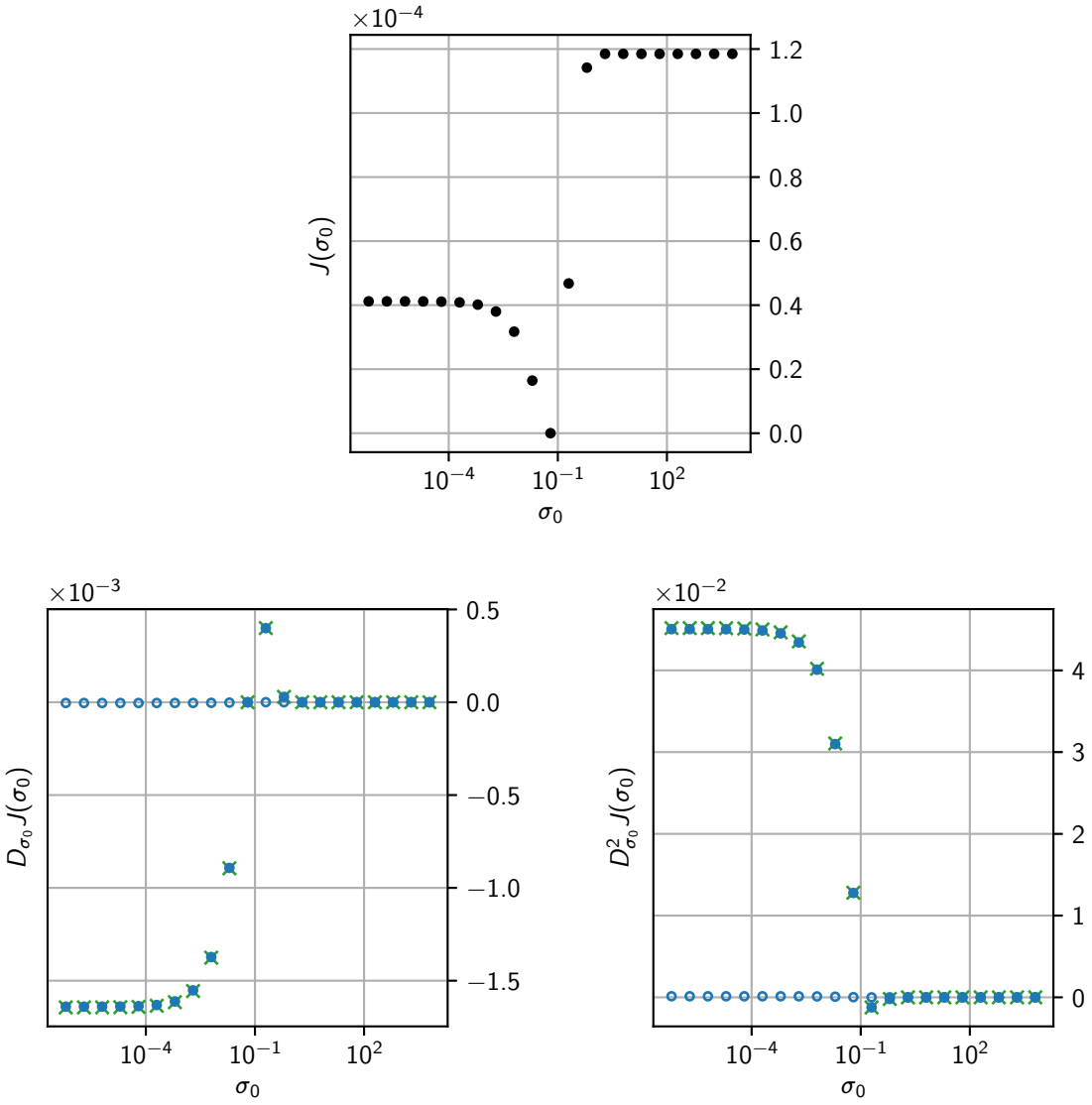


Figure B.12: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\epsilon}_c = 1.2$, $\tilde{\sigma}_c = \pi \times 10^{-1}$) embedded in a background medium whose conductivity σ_0 varies around the ground-truth value $\tilde{\sigma}_0 = 2\pi \times 10^{-2}$ while the permittivity is kept at the ground-truth value $\tilde{\epsilon}_0 = 1.2$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

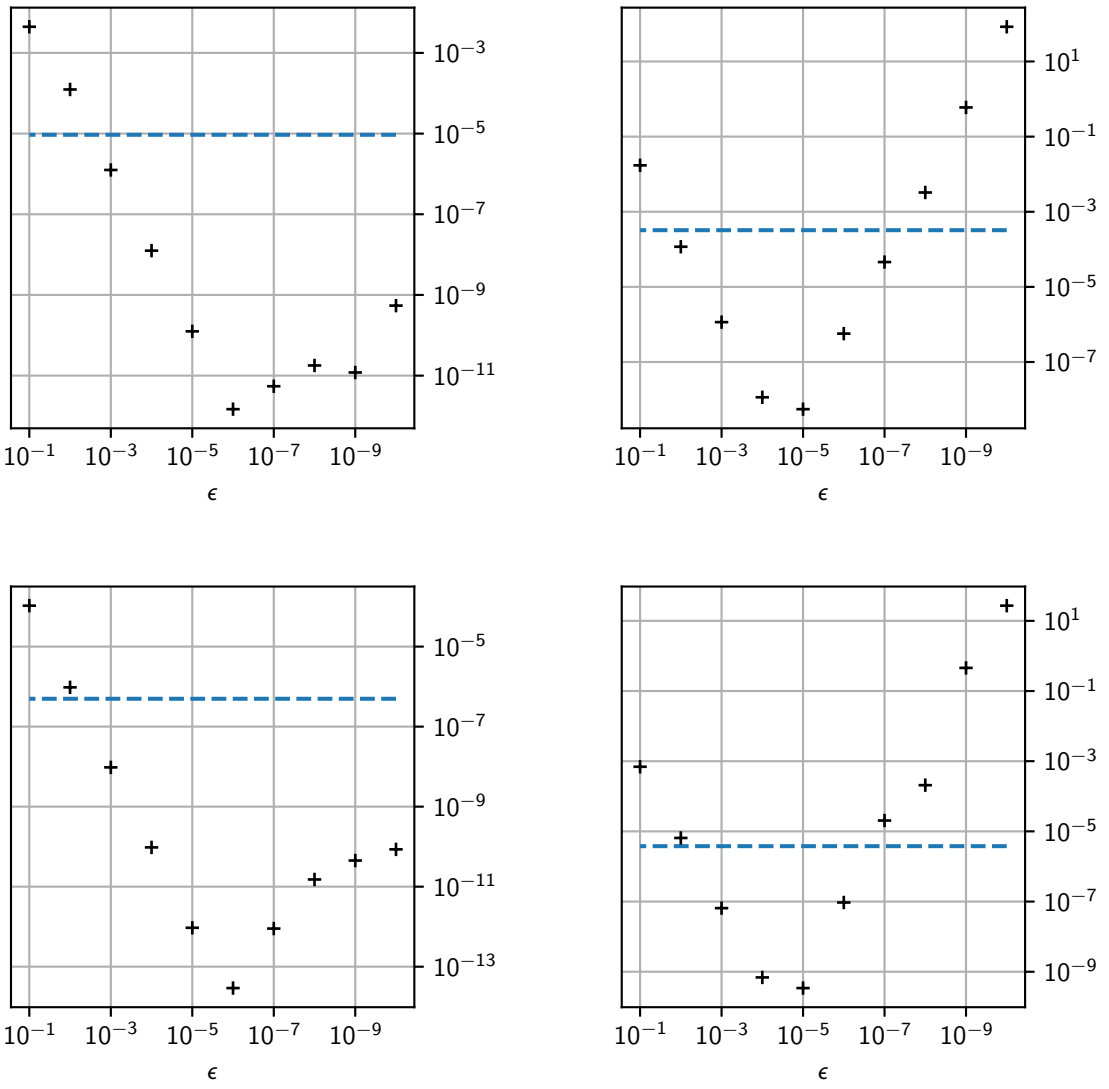


Figure B.13: Difference between the approximation of (1.3) and the adjoint state method with the boundary contribution for a highly penetrable cylinder ($\tilde{\epsilon}_c = 1.2$, $\tilde{\sigma}_c = \pi \times 10^{-1}$) when the unknown background permittivity and conductivity are respectively $\epsilon_0 = 1.1$ and $\sigma_0 = 6\pi \times 10^{-2}$ while the ground-truth value are respectively $\tilde{\epsilon}_0 = 1.0$ and $\tilde{\sigma}_0 = 2\pi \times 10^{-2}$. Top row is related to permittivity derivatives, bottom row to conductivity derivatives. Left column is related to first order derivatives, right column to second order derivatives. The dashed blue line (−−) is the amplitude of the boundary contribution in the adjoint state method.

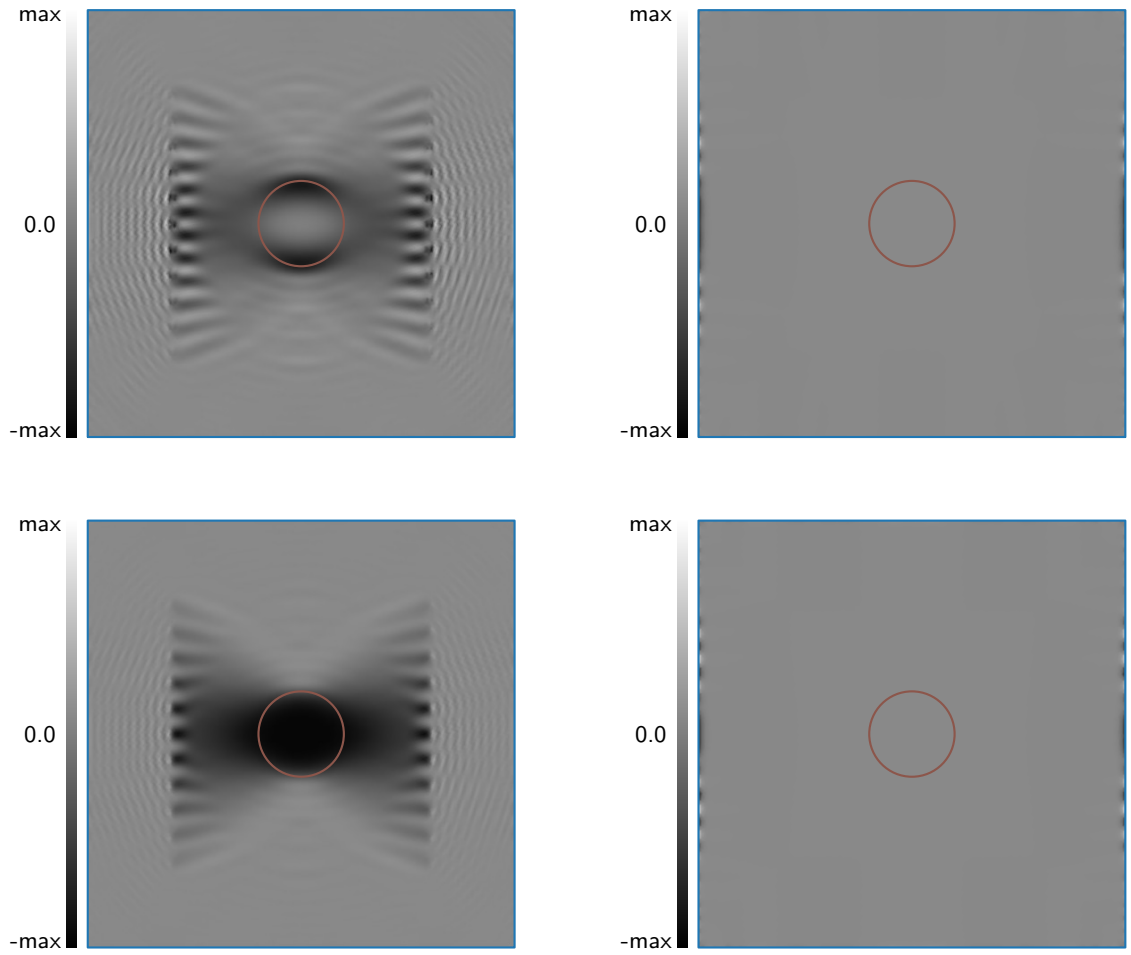


Figure B.14: Bulk (left column) and boundary (right column) parts of the H_1-Sobolev gradient ($l_c := 2\pi\sqrt{\alpha_1}$; $l_c = 1$). The data used for this gradient is obtained with a highly penetrable cylinder ($\tilde{\epsilon}_c = 1.2$, $\tilde{\epsilon}_0 = 1.0$ and $\tilde{\sigma}_c = \pi \times 10^{-1}$, $\ll \tilde{sigma}_0 = 2\pi \times 10^{-2}$). Top row is permittivity gradients. Bottom row is conductivity gradients. From top to bottom, the maximal value for the bulk and boundary contribution are respectively 15×10^{-3}, 15×10^{-3} (bulk) and 3×10^{-3}, 3×10^{-3} (boundary).

B.3 Navier

There are two types of elastic waves: shear waves and pressure waves. Consequently, two impedance parameters β_s and β_p must be used in the absorbing boundary condition to absorb both type of waves

$$\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \hat{\boldsymbol{n}} + \beta_{s,0}(\lambda, \mu, \rho)\gamma_T(\boldsymbol{u}) + \beta_{p,0}(\lambda, \mu, \rho)\gamma_n(\boldsymbol{u}) = 0. \tag{B.23}$$

The systematic steps of the adjoint state method are as follows, in terms of three bulk model parameters ρ, λ and μ and two boundary model parameter β_s and β_p:

1. Find the direct states $\boldsymbol{u}_e$ obeying the direct problem

$$\begin{cases} \operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u}_e)) + \omega^2 \rho \boldsymbol{u}_e & = \hat{\boldsymbol{f}}\delta(\boldsymbol{x} - \boldsymbol{x}_e) \quad \text{in } \Omega \\ \boldsymbol{\sigma}(\boldsymbol{u}_e) \cdot \hat{\boldsymbol{n}} + \beta_s \gamma_T(\boldsymbol{u}_e) + \beta_p \gamma_n(\boldsymbol{u}_e) & = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{B.24}$$

2. Find the adjoint states $\overline{\boldsymbol{u}}_e^\dagger$ obeying the adjoint problem

$$\begin{cases} \operatorname{div}\left(\boldsymbol{\sigma}(\overline{\boldsymbol{u}}_e^\dagger)\right) + \omega^2 \rho \overline{\boldsymbol{u}}_e^\dagger & = \sum_r (\overline{\boldsymbol{u}}_e(\boldsymbol{x}_r) - \overline{d}_{e,r})\delta(\boldsymbol{x} - \boldsymbol{x}_r) \quad \text{in } \Omega \\ \boldsymbol{\sigma}(\overline{\boldsymbol{u}}_e^\dagger) \cdot \hat{\boldsymbol{n}} + \beta_s \gamma_T(\overline{\boldsymbol{u}}_e^\dagger) + \beta_p \gamma_n(\overline{\boldsymbol{u}}_e^\dagger) & = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{B.25}$$

3. Once the direct and adjoint states are known, the gradient kernels with respect to the different model parameters can be evaluated:

   - bulk sensitivity w.r.t. ρ

$$\vec{j}'_\Omega = \sum_e \operatorname{div}\left(\overline{\boldsymbol{u}}_e^\dagger\right) \operatorname{div}(\boldsymbol{u}_e) \tag{B.26}$$

   - bulk sensitivity w.r.t. λ

$$\vec{j}'_\Omega = 2 \sum_e \boldsymbol{\epsilon}(\overline{\boldsymbol{u}}_e^\dagger) : \boldsymbol{\epsilon}(\boldsymbol{u}_e) \tag{B.27}$$

   - bulk sensitivity w.r.t. μ

$$\vec{j}'_\Omega = -\omega^2 \sum_e \overline{\boldsymbol{u}}_e^\dagger \cdot \boldsymbol{u}_e \tag{B.28}$$

   - boundary sensitivity w.r.t. β_s

$$\vec{j}'_{\partial\Omega} = \sum_e \gamma_T(\overline{\boldsymbol{u}}_e^\dagger) \cdot \gamma_T(\boldsymbol{u}_e) \tag{B.29}$$

- boundary sensitivity w.r.t. β_p

$$\vec{J}'_{\partial\Omega} = \sum_e \gamma_n(\overline{\boldsymbol{u}}_e^\dagger) \cdot \gamma_n(\boldsymbol{u}_e). \tag{B.30}$$

4. Find the perturbed direct states $\delta\boldsymbol{u}_e$ obeying the direct problem

$$\begin{cases} \operatorname{div}(\boldsymbol{\sigma}(\delta\boldsymbol{u}_e)) + \omega^2\rho\delta\boldsymbol{u}_e & = -\operatorname{div}(\delta\boldsymbol{\sigma}(\boldsymbol{u}_e)) + \omega^2\delta\rho\boldsymbol{u}_e & \text{in } \Omega \\ \boldsymbol{\sigma}(\delta\boldsymbol{u}_e)\cdot\hat{\boldsymbol{n}} + \beta_s\gamma_T(\delta\boldsymbol{u}_e) + \beta_p\gamma_n(\delta\boldsymbol{u}_e) & = -\boldsymbol{\delta\sigma}(\boldsymbol{u}_e)\cdot\hat{\boldsymbol{n}} & \\ & \quad + \delta\beta_s\gamma_T(\boldsymbol{u}_e) + \delta\beta_p\gamma_n(\boldsymbol{u}_e) & \text{on } \partial\Omega. \end{cases} \tag{B.31}$$

5. Find the perturbed adjoint states $\delta\overline{\boldsymbol{u}}_e^\dagger$ obeying the adjoint problem

$$\begin{cases} \operatorname{div}\left(\boldsymbol{\sigma}(\delta\overline{\boldsymbol{u}}_e^\dagger)\right) + \omega^2\rho\delta\overline{\boldsymbol{u}}_e^\dagger & = \sum_r \delta\overline{\boldsymbol{u}}_e(\boldsymbol{x}_r)\delta(\boldsymbol{x}-\boldsymbol{x}_r) & \\ & \quad - \operatorname{div}\left(\delta\boldsymbol{\sigma}(\overline{\boldsymbol{u}}_e^\dagger)\right) + \omega^2\delta\rho\overline{\boldsymbol{u}}_e^\dagger & \text{in } \Omega \\ \boldsymbol{\sigma}(\overline{\boldsymbol{u}}_e^\dagger)\cdot\hat{\boldsymbol{n}} + \beta_s\gamma_T(\overline{\boldsymbol{u}}_e^\dagger) + \beta_p\gamma_n(\overline{\boldsymbol{u}}_e^\dagger) & = -\boldsymbol{\delta\sigma}(\overline{\boldsymbol{u}}_e^\dagger)\cdot\hat{\boldsymbol{n}} & \\ & \quad + \delta\beta_s\gamma_T(\overline{\boldsymbol{u}}_e^\dagger) + \delta\beta_p\gamma_n(\overline{\boldsymbol{u}}_e^\dagger) & \text{on } \partial\Omega. \end{cases} \tag{B.32}$$

6. Once the perturbed direct and adjoint states are known, the application of the Hessian operator in a particular direction can be evaluated:

- bulk second order sensitivity w.r.t. ρ

$$\delta\vec{j}'_\Omega = \sum_e \operatorname{div}\left(\delta\overline{\boldsymbol{u}}_e^\dagger\right)\operatorname{div}(\boldsymbol{u}_e) + \operatorname{div}\left(\overline{\boldsymbol{u}}_e^\dagger\right)\operatorname{div}(\delta\boldsymbol{u}_e) \tag{B.33}$$

- bulk second order sensitivity w.r.t. λ

$$\delta\vec{j}'_\Omega = 2\sum_e \boldsymbol{\epsilon}(\delta\overline{\boldsymbol{u}}_e^\dagger) : \boldsymbol{\epsilon}(\boldsymbol{u}_e) + \boldsymbol{\epsilon}(\overline{\boldsymbol{u}}_e^\dagger) : \boldsymbol{\epsilon}(\delta\boldsymbol{u}_e) \tag{B.34}$$

- bulk second order sensitivity w.r.t. μ

$$\delta\vec{j}'_\Omega = -\omega^2\sum_e \delta\overline{\boldsymbol{u}}_e^\dagger\cdot\boldsymbol{u}_e + \overline{\boldsymbol{u}}_e^\dagger\cdot\delta\boldsymbol{u}_e \tag{B.35}$$

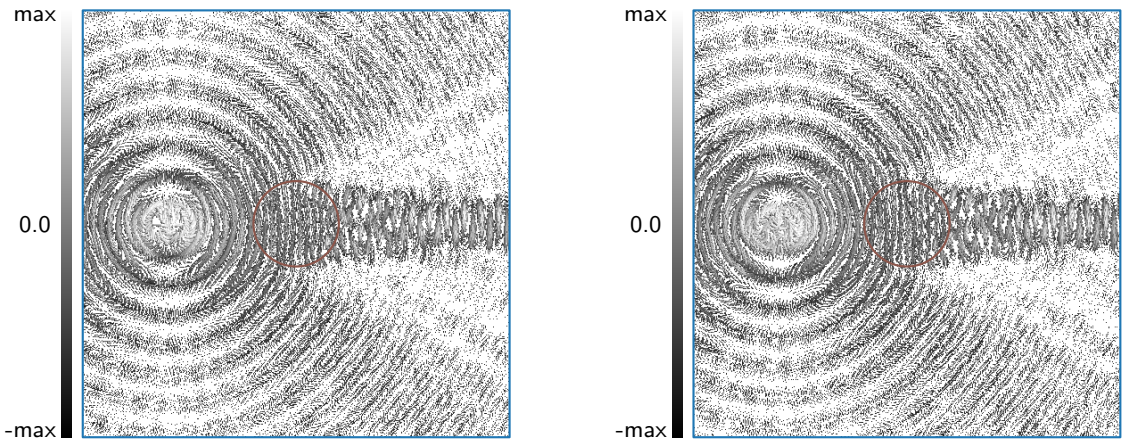
- boundary second order sensitivity w.r.t. β_s

$$\vec{J}'_{\partial\Omega} = \sum_e \gamma_T(\delta\overline{\boldsymbol{u}}_e^\dagger)\cdot\gamma_T(\boldsymbol{u}_e) + \gamma_T(\overline{\boldsymbol{u}}_e^\dagger)\cdot\gamma_T(\delta\boldsymbol{u}_e) \tag{B.36}$$

- boundary second order sensitivity w.r.t. β_p

$$\vec{J}'_{\partial\Omega} = \sum_e \gamma_n(\delta\overline{\boldsymbol{u}}_e^\dagger) \cdot \gamma_n(\boldsymbol{u}_e) + \gamma_n(\overline{\boldsymbol{u}}_e^\dagger) \cdot \gamma_n(\delta\boldsymbol{u}_e). \tag{B.37}$$

For simplicity, the unbounded propagation domain is again modeled by means of a zeroth order absorbing condition, for which the relationships $\beta_{s,0}(\lambda, \mu, \rho) := i\omega\sqrt{\rho\mu}$ and $\beta_{p,0} := i\omega\sqrt{\rho(\lambda + 2\mu)}$ hold. Derivatives w.r.t. ρ, λ or μ can easily be obtained from the derivatives w.r.t. β_s and β_p through the chain rule. For the numerical illustrations, the excitation is polarized along the y-axis, *i.e.* $\hat{\boldsymbol{f}} = \hat{\boldsymbol{y}}$, and only the y-component of the displacement field is measured.



(a) Re $(\boldsymbol{u})$ (b) Im $(\boldsymbol{u})$

Figure B.15: Ground-truth direct field $\boldsymbol{u}$ for a highly penetrable cylinder and for a single source. The background and cylinder permittivity and conductivity are respectively $\tilde{\rho}_c = 1.2$, $\tilde{\lambda}_c = 0.8$, $\tilde{\mu}_c = 0.8$ and $\tilde{\rho}_0 = 1.$, $\tilde{\lambda}_0 = 1.$, $\tilde{\mu}_0 = 1.$.

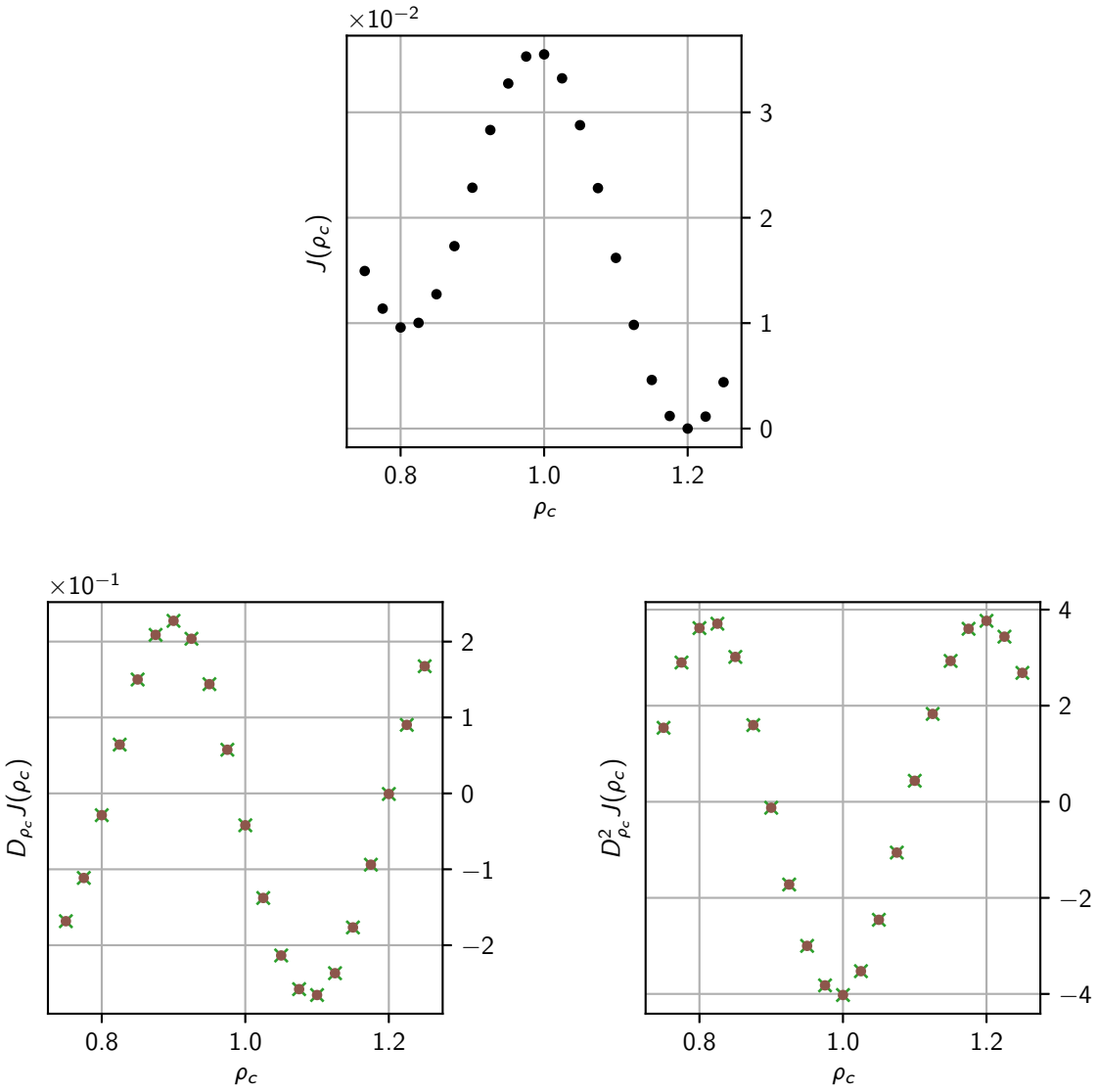


Figure B.16: Performance functional (•) and its derivatives for a highly penetrable cylinder, embedded in a background medium ($\tilde{\rho}_0 = 1.$, $\tilde{\lambda}_0 = 1.$, $\tilde{\mu}_0 = 1.$), whose density ρ_c varies around the ground-truth value $\tilde{\rho}_c = 1.2$ while Lamé parameters are kept at the ground-truth value $\tilde{\lambda}_c = 0.8$ and $\tilde{\mu}_c = 0.8$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

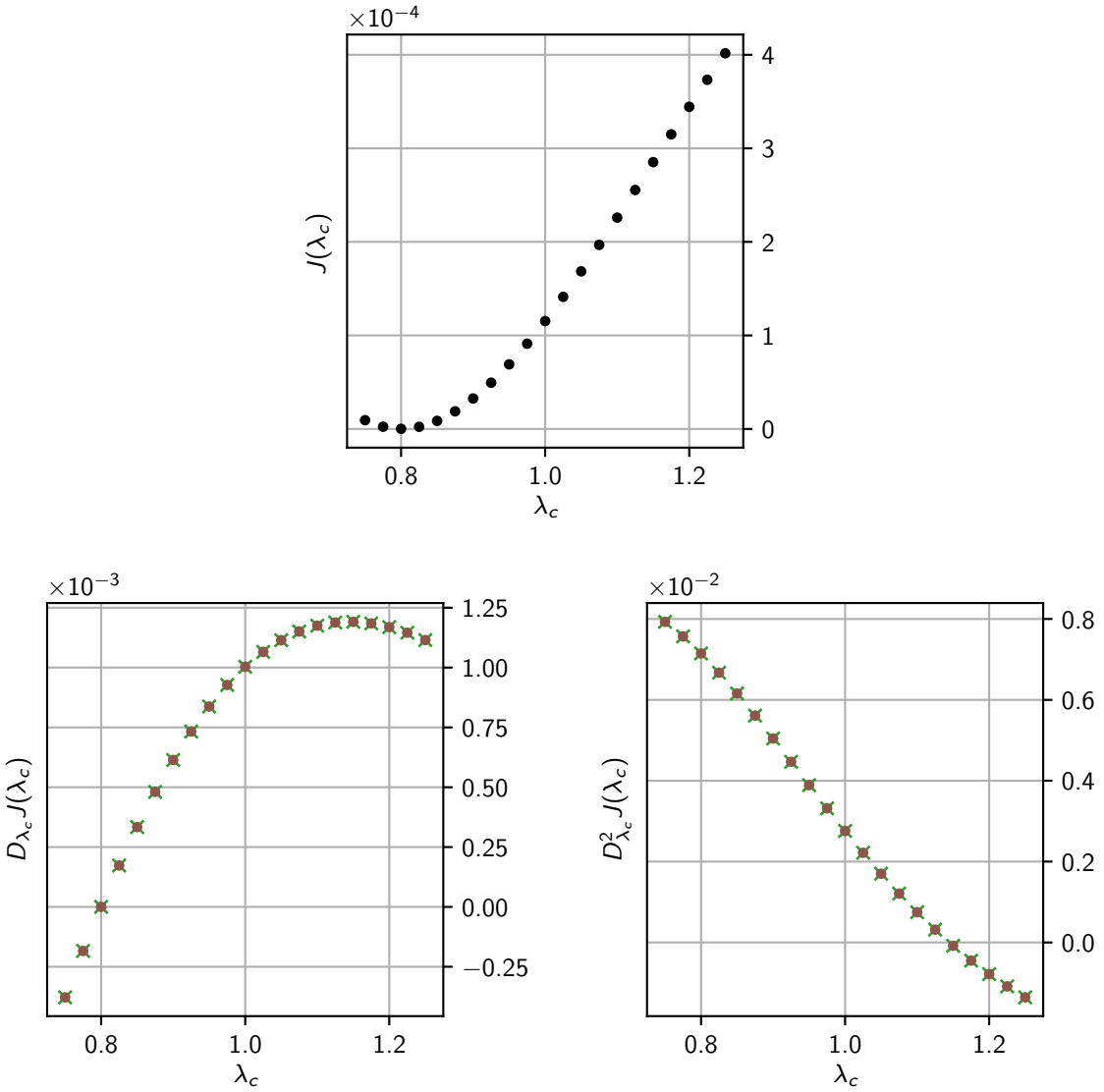


Figure B.17: Performance functional (•) and its derivatives for a highly penetrable cylinder, embedded in a background medium ($\tilde{\rho}_0 = 1.$, $\tilde{\lambda}_0 = 1.$, $\tilde{\mu}_0 = 1.$), whose first Lamé parameter λ_c varies around the ground-truth value $\tilde{\lambda}_c = 0.8$ while the density and the second Lamé parameter are kept at the ground-truth value $\tilde{\rho}_c = 1.2$ and $\tilde{\mu}_c = 0.8$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

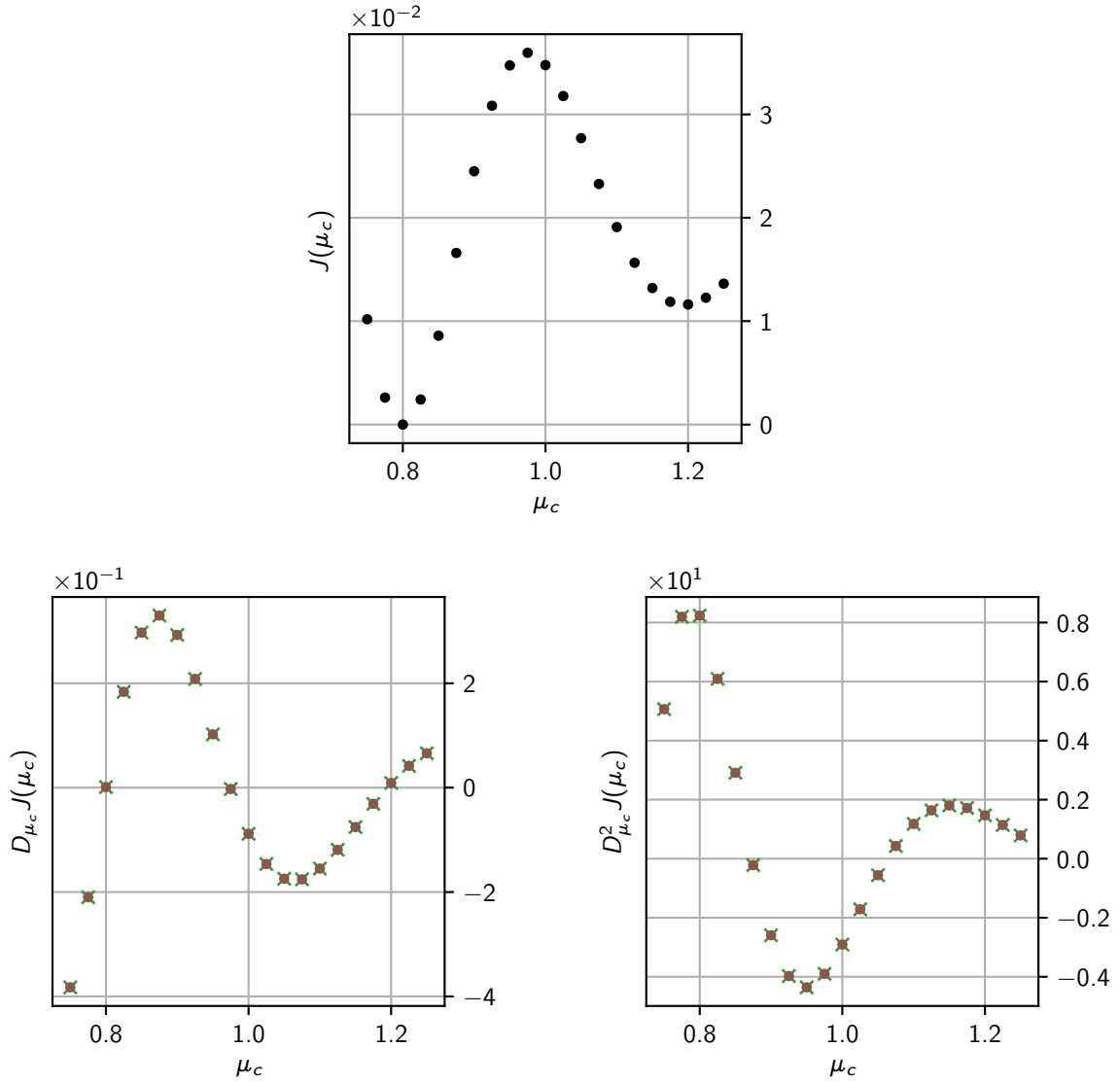


Figure B.18: Performance functional (•) and its derivatives for a highly penetrable cylinder, embedded in a background medium ($\tilde{\rho}_0 = 1.$, $\tilde{\lambda}_0 = 1.$, $\tilde{\mu}_0 = 1.$), whose second Lamé parameter μ_c varies around the ground-truth value $\tilde{\mu}_c = 0.8$ while the density and the second Lamé parameter are kept at the ground-truth value $\tilde{\rho}_c = 1.2$ and $\tilde{\lambda}_c = 0.8$. The derivatives are computed by the adjoint state method (•) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×).

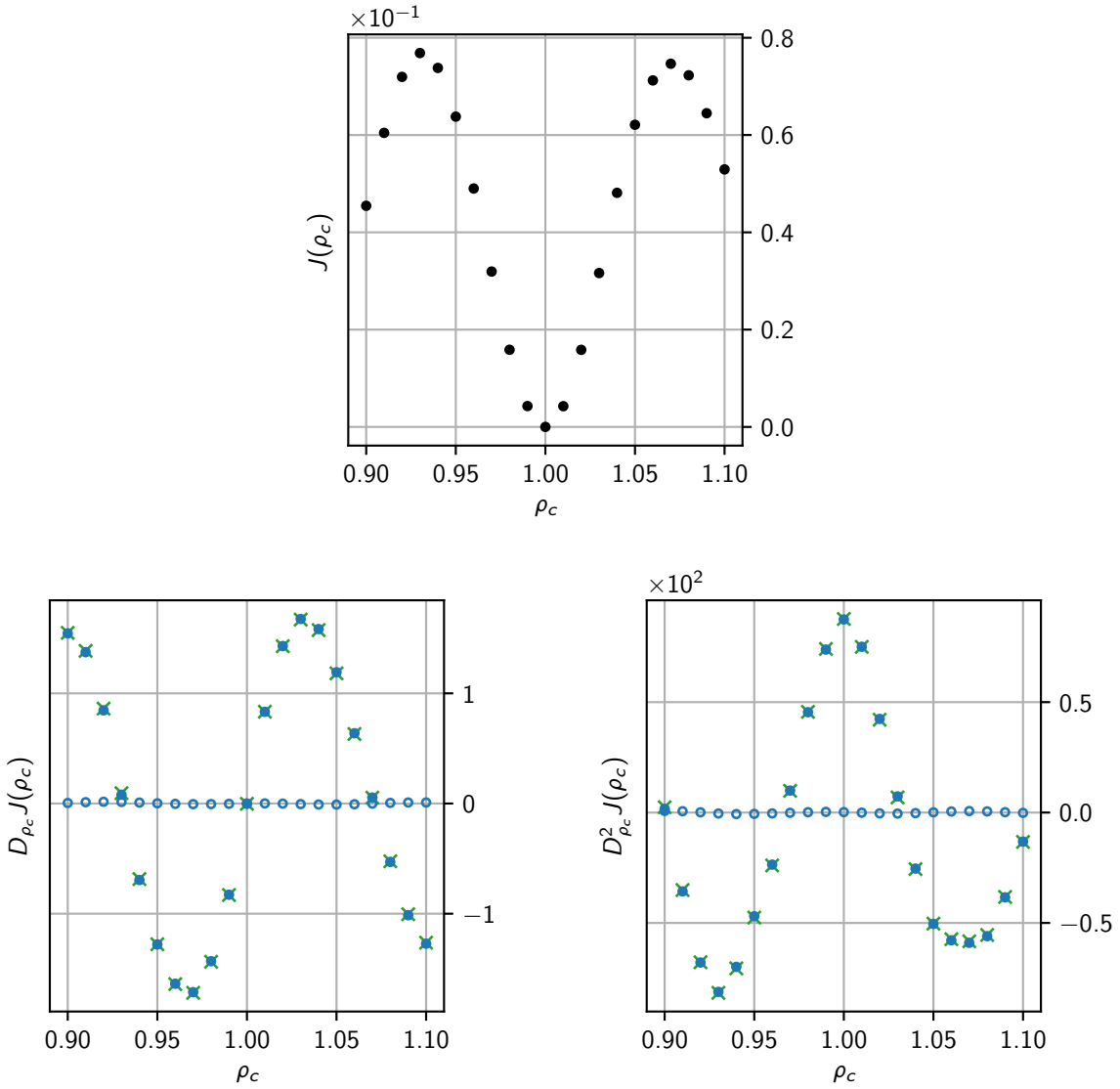


Figure B.19: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\rho}_c = 1.2$, $\tilde{\lambda}_c = 0.8$, $\tilde{\mu}_c = 0.8$) embedded in a background medium whose density ρ_0 varies around the ground-truth value $\tilde{\rho}_0 = 1.0$ while the Lamé parameters are kept at the ground-truth value $\tilde{\lambda}_0 = 1.0$ and $\tilde{\mu}_0 = 1.0$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

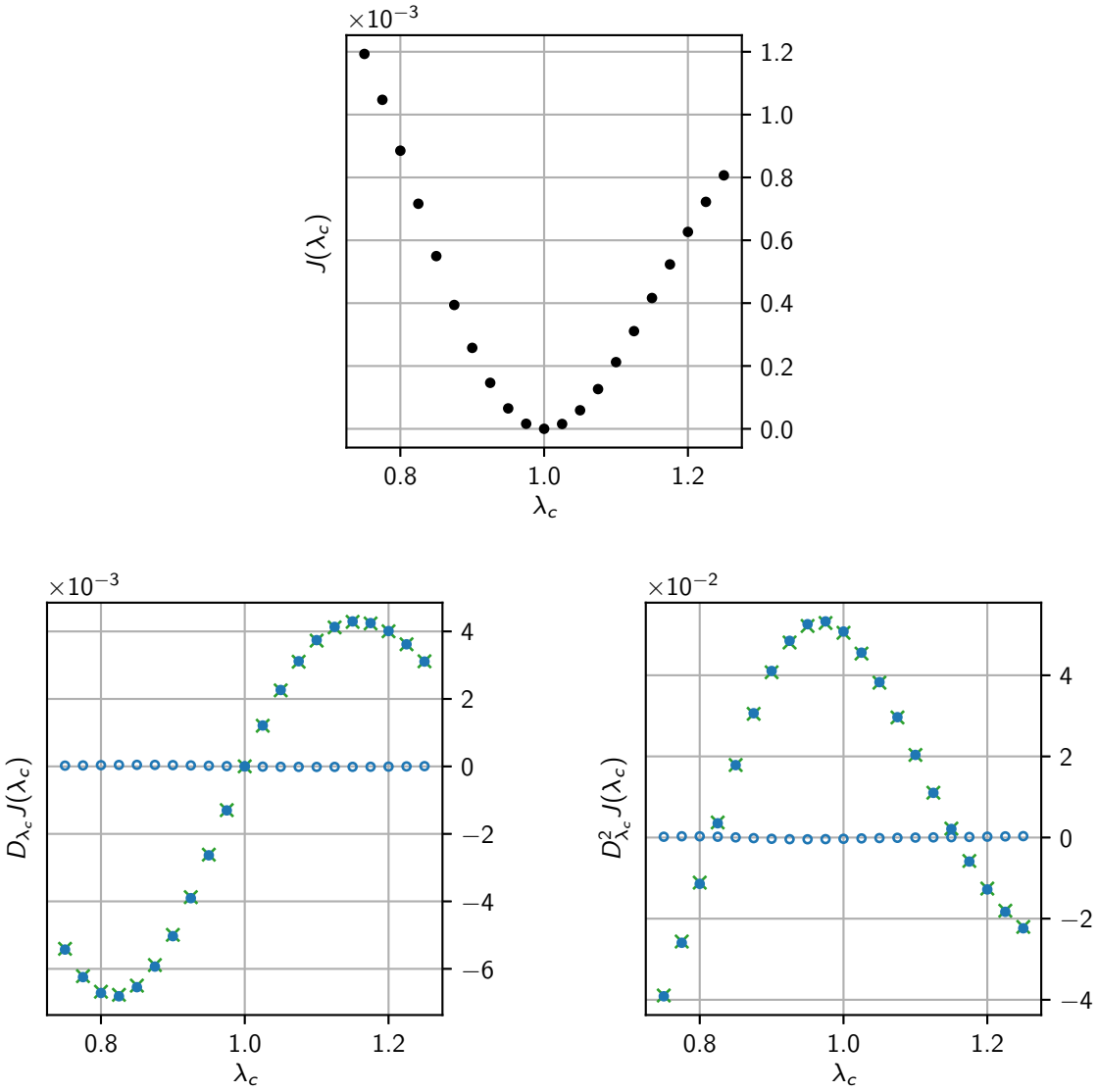


Figure B.20: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\rho}_c = 1.2$, $\tilde{\lambda}_c = 0.8$, $\tilde{\mu}_c = 0.8$) embedded in a background medium whose first Lamé parameters λ_0 varies around the ground-truth value $\tilde{\lambda}_0 = 1.0$ while the density and the second Lamé parameters are kept at the ground-truth value $\tilde{\rho}_0 = 1.0$ and $\tilde{\mu}_0 = 1.0$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

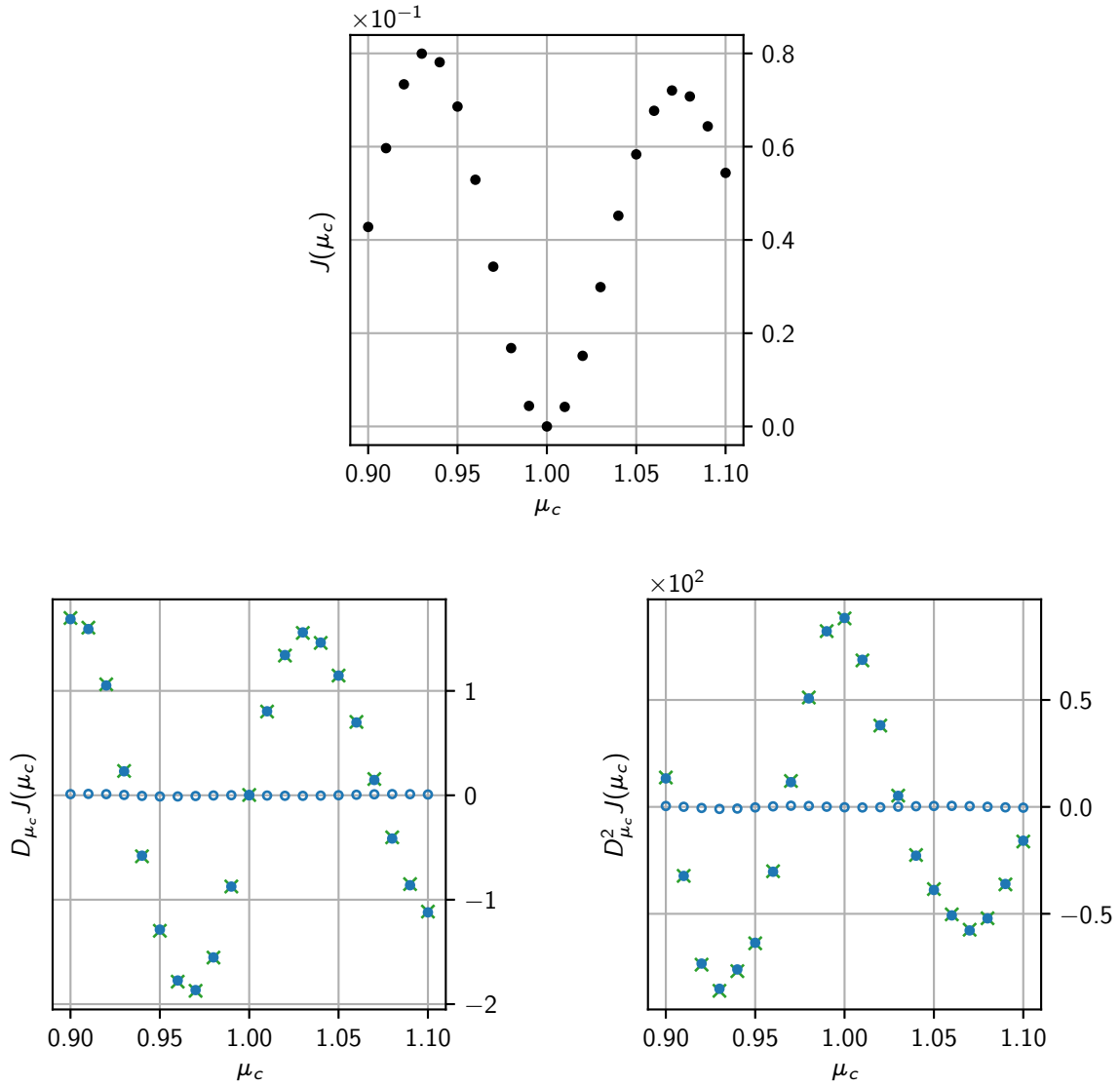


Figure B.21: Performance functional (•) and its derivatives for a highly penetrable cylinder ($\tilde{\rho}_c = 1.2$, $\tilde{\lambda}_c = 0.8$, $\tilde{\mu}_c = 0.8$) embedded in a background medium whose second Lamé parameters μ_0 varies around the ground-truth value $\tilde{\mu}_0 = 1.0$ while the density and the first Lamé parameters are kept at the ground-truth value $\tilde{\rho}_0 = 1.0$ and $\tilde{\lambda}_0 = 1.0$. The derivatives are computed by the adjoint state method (•, ○) and by an approximation of the definition (1.3) (with $\epsilon = 10^{-5}$) (×). The bulk (•) and boundary (○) contributions of the adjoint state method are plotted separately.

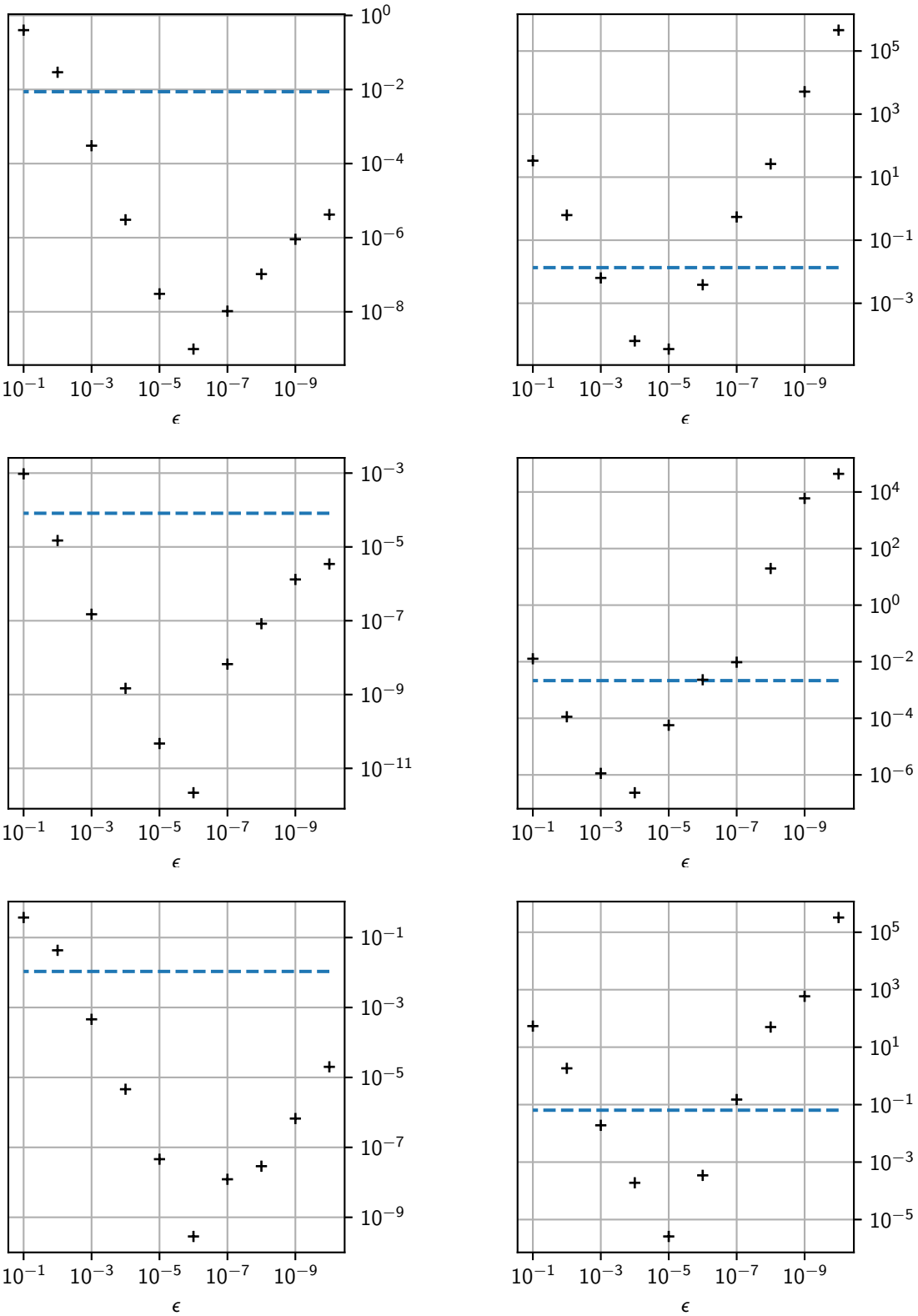


Figure B.22: Difference between the approximation of (1.3) and the adjoint state method with the boundary contribution for a highly penetrable cylinder ($\tilde{\rho}_c = 1.2$, $\tilde{\lambda}_c = 0.8$, $\tilde{\mu}_c = 0.8$) when the unknown background density and Lamé parameters are respectively $\rho_0 = 1.1$, $\lambda_0 = 0.9$ and $\mu_0 = 0.9$ while the ground-truth value are respectively $\tilde{\rho}_0 = 1.0$, $\tilde{\lambda}_0 = 1.0$ and $\tilde{\mu}_0 = 1.0$. Top row is related to density derivatives, middle and bottom rows respectively to first and second Lamé parameters derivatives: first derivatives are the left column, second derivatives are the right column. The dashed blue line (−−) is the amplitude of the boundary contribution in the adjoint state method.

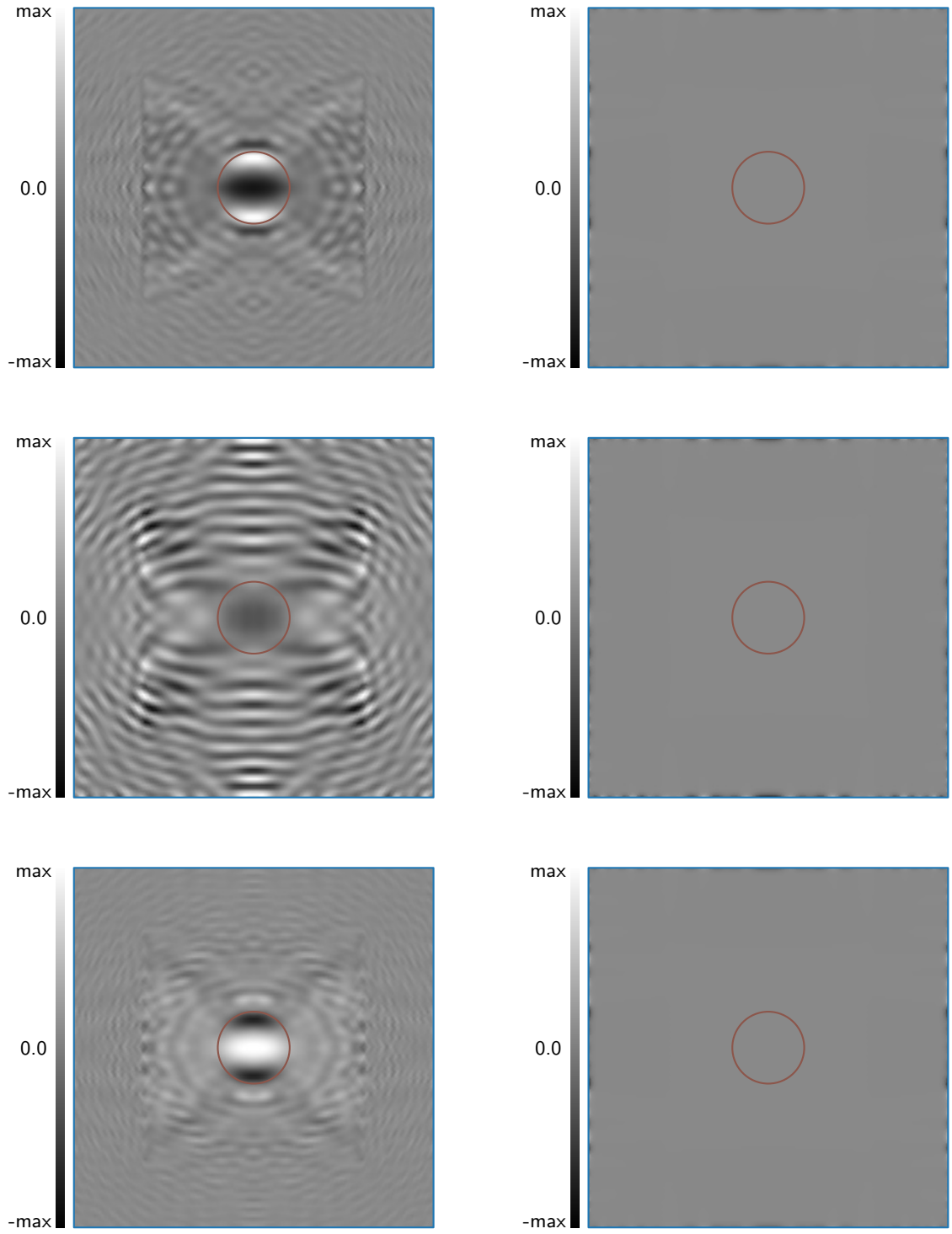


Figure B.23: Bulk (left column) and boundary (right column) parts of the H_1-Sobolev gradient ($l_c := 2\pi\sqrt{\alpha_1}$; $l_c = 1$). The data used for this gradient is obtained with a highly penetrable cylinder ($\tilde{\rho}_0 = 1.0$, $\tilde{\lambda}_0 = 1.0$, $\tilde{\mu}_0 = 1.0$ and $\tilde{\rho}_c = 1.2$, $\tilde{\lambda}_c = 0.8$, $\tilde{\mu}_c = 0.8$). Top row is density gradients. Middle row is Lamé first parameter gradients. Bottom row is Lamé second parameter gradients. From top to bottom, the maximal value for the bulk and boundary contribution are respectively 8×10^{-3}, 15×10^{-5}, 7×10^{-3} (bulk) and 16×10^{-4}, 23×10^{-5}, 15×10^{-4} (boundary).

Appendix C

Gauss-Newton Hessian operator

The *Gauss-Newton* Hessian operator is a positive definite approximation of the full Hessian operator which can be defined when the misfit functional consists in a sum of some squared residuals $Q_q(m)$, *i.e.*

$$J(m) := \frac{1}{2}\sum_q |Q_q(m)|^2 , \tag{C.1}$$

as for example the least squares distance

$$Q_q(m) = u_q(m) - d_q. \tag{C.2}$$

In that specific case, the second order directional derivatives, from which the Hessian operator originates, is composed of two terms

$$\begin{aligned}\{D^2_{mm}J(m)\}(\delta m_1, \delta m_2) = \mathrm{Re}\sum_q \{D_m\overline{Q}_q(m)\}(\delta m_1)\{D_m Q_q(m)\}(\delta m_2) \\ + \mathrm{Re}\sum_q \overline{Q}_q(m)\{D^2_{mm}Q_q(m)\}(\delta m_1, \delta m_2).\end{aligned} \tag{C.3}$$

The first term is clearly positive semi-definite: it is positive or null for $\delta m_1 = \delta m_2$, but the second term is indefinite. The idea behind the Gauss-Newton approximation in order to obtain a positive semi-definite operator is therefore to neglect this second term. This approximation is thus valid if the residuals are small or linear w.r.t. the model parameters m. In the vicinity of the global minimum, these residuals are indeed nearly vanishing but this is not necessarily true elsewhere. When the residuals are not small, the validity of the approximation then relies on the (non-)linearity of the residuals. In the context of the least squared distance (or actually any residuals that depends on m through a linear term in $u_q(m)$), this second order residuals derivative equals (or is proportional to)

$$\{D^2_{mm}Q_q(m)\}(\delta m_1, \delta m_2) = \delta_{1,2}u. \tag{C.4}$$

In other words, it is proportional to a second order perturbed wavefield. Intuitively, the first order perturbed wavefield $\delta_1 u$ is related to the single scattering provoked by a small perturbation δm_1 of the model parameter and the second order perturbed wavefield $\delta_{1,2} u$ is therefore related to perturbation of this single scattering response due to a second small perturbation δm_2. Hence it is related to doubly scattered wavefield.
In summary, in the context of full waveform inversion optimization algorithm, this approximation is valid close to the global minimum or in the absence of strong multiple scattering.
As far as practical computation are concerned, the same procedure as for the full Newton Hessian can be followed. The only difference is that, after the computation of the gradient, the residuals should be set to zero for the remainder of the procedure, *i.e.* for steps 5 and 6. As an example, the adjoint procedure of Chapter 1 for Helmholtz (I) equation is adapted here below with the Gauss-Newton approximation. Boundary terms are omitted for compactness.

1. Find the direct states u_e obeying the direct problem

$$\operatorname{div}\left(\mathbf{grad}\left(u_e\right)\right) + \omega^2 s^2 u_e \quad = \delta(\boldsymbol{x} - \boldsymbol{x}_e) \quad \text{in } \Omega. \tag{C.5}$$

2. Find the adjoint states $u_e^\dagger$ obeying the adjoint problem

$$\operatorname{div}\left(\mathbf{grad}\left(\overline{u}_e^\dagger\right)\right) + \omega^2 s^2 \overline{u}_e^\dagger \quad = \sum_r (\overline{u}_e(\boldsymbol{x}_r) - \overline{d}_{e,r})\delta(\boldsymbol{x} - \boldsymbol{x}_r) \quad \text{in } \Omega. \tag{C.6}$$

3. Once the direct and adjoint states are known, the associated gradient kernel can be identified:

$$\overline{j}' = -\omega^2 \sum_e u_e \overline{u}_e^\dagger. \tag{C.7}$$

4. Find the perturbed direct states δu_e obeying the perturbed direct problem

$$\left\{\operatorname{div}\left(\mathbf{grad}\left(\delta u_e\right)\right) + \omega^2 s^2 \delta u_e \quad = -\omega^2 \delta s^2 u_e \quad \text{in } \Omega. \right. \tag{C.8}$$

5. Find the perturbed adjoint states $\delta u_e^\dagger$ obeying the perturbed adjoint problem

$$\operatorname{div}\left(\mathbf{grad}\left(\delta \overline{u}_e^\dagger\right)\right) + \omega^2 s^2 \delta \overline{u}_e^\dagger \quad = \sum_r \delta \overline{u}_e(\boldsymbol{x}_r)\delta(\boldsymbol{x} - \boldsymbol{x}_r) - \cancel{\omega^2 \delta s^2 \overline{u}_e^\dagger} \quad \text{in } \Omega. \tag{C.9}$$

   The adjoint wavefield $\overline{u}^\dagger$ is neglected from here because it is linearly proportional to the residuals. Interestingly, the remaining source term as a very localized support, *i.e.* the point receivers. It is thus very quick to assemble.

6. Once the perturbed direct and adjoint states are known, the application of the Gauss-

Newton Hessian operator in a particular direction can be evaluated:

$$\delta\overline{j'} = -\omega^2 \sum_e \cancel{\delta u_e \overline{u}_e^\dagger} + u_e \delta\overline{u}_e^\dagger. \tag{C.10}$$

C.1 Diagonal approximation

In the context of the Helmholtz I equation, the forward, adjoint, perturbed forward and perturbed adjoint wave fields appearing in the adjoint state method can be expressed analytically in terms of Green functions $g(\boldsymbol{x}, \boldsymbol{y})$ using the superposition principle. From (C.5), (C.6), (C.8) and (C.9) respectively, it comes

$$\begin{aligned} u_e(\boldsymbol{x}) &= g(\boldsymbol{x}, \boldsymbol{x}_e), & \overline{u}_e^\dagger(\boldsymbol{x}) &= \sum_r (\overline{u}_e(\boldsymbol{x}_r) - \overline{d}_{e,r}) g(\boldsymbol{x}_r, \boldsymbol{x}), \\ \delta\overline{u}_e^\dagger &= \sum_r \delta\overline{u}_e(\boldsymbol{x}_r) g(\boldsymbol{x}_r, \boldsymbol{x}), & \delta u_e(\boldsymbol{x}) &= -\omega^2 \int_\Omega g(\boldsymbol{x}, \boldsymbol{y}) u_e(\boldsymbol{y}) \delta s^2(\boldsymbol{y})\, d\boldsymbol{y}. \end{aligned} \tag{C.11}$$

Using the above wavefields expressions, the conventional Gauss-Newton Hessian operator in the direction δs^2 is successively given by

$$H_{\mathrm{GN}} \delta s^2 = -\omega^2 \sum_e \overline{u}_e \delta u_e^\dagger = -\omega^2 \sum_e \overline{g}(\boldsymbol{x}, \boldsymbol{x}_e) \sum_r \delta u_e(\boldsymbol{x}_r) \overline{g}(\boldsymbol{x}_r, \boldsymbol{x}) \tag{C.12}$$

$$= \omega^4 \sum_{e,r} \overline{g}(\boldsymbol{x}, \boldsymbol{x}_e) \overline{g}(\boldsymbol{x}_r, \boldsymbol{x}) \int_\Omega g(\boldsymbol{x}_r, \boldsymbol{y}) u_e(\boldsymbol{y}) \delta s^2(\boldsymbol{y})\, d\boldsymbol{y} \tag{C.13}$$

$$:= \int_\Omega h_{\mathrm{GN}}(x, y) \delta s^2(\boldsymbol{y})\, d\boldsymbol{y} \tag{C.14}$$

with the Gauss-Newton Hessian kernel

$$h_{\mathrm{GN}}(\boldsymbol{x}, \boldsymbol{y}) = \omega^4 \sum_{e,r} g(\boldsymbol{y}, \boldsymbol{x}_e) \overline{g}(\boldsymbol{x}, \boldsymbol{x}_e)\, g(\boldsymbol{x}_r, \boldsymbol{y}) \overline{g}(\boldsymbol{x}_r, \boldsymbol{x}). \tag{C.15}$$

The diagonal part of the Gauss-Newton Hessian operator is then finally identified as

$$\mathrm{diag}\,(H_{\mathrm{GN}}) = h_{\mathrm{GN}}(\boldsymbol{x}, \boldsymbol{x})\, A = \omega^4 \sum_{e,r} |g(\boldsymbol{x}, \boldsymbol{x}_e)|^2 \; |g(\boldsymbol{x}_r, \boldsymbol{x})|^2 \; A. \tag{C.16}$$

The area A is related to the support of integral (C.14). It is thus of the order of the wavelength squared [144]. With these conventions, the application of the diagonal operator in the direction δs^2 is then $\mathrm{diag}\,(H_{\mathrm{GN}})\, \delta s^2 = \mathrm{diag}\,(H_{\mathrm{GN}})\,(\boldsymbol{x})\, \delta s^2(\boldsymbol{x})$. Detail about the Gauss-Newton Hessian can be found in [54, 55, 117, 118, 135].

Appendix D

Permittivity/Conductivity evolution during an inversion

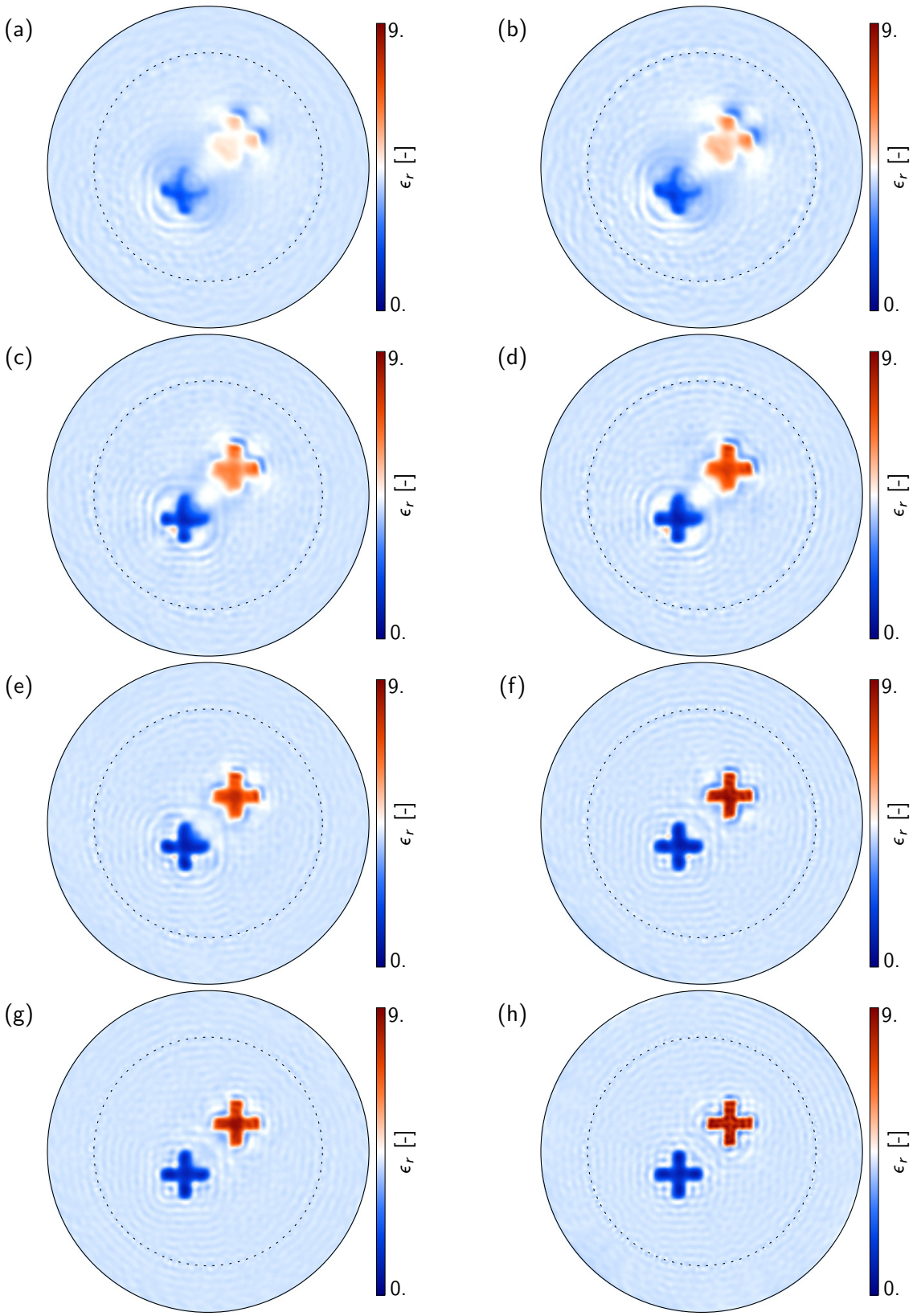


Figure D.1: Intermediate inversion results (permittivity) for the line search *l*-BFGS algorithm with two different scaling for the inner product: $\beta = 2$ (left) and $\beta = 5$ (right). From top to bottom, these reconstruction corresponds to the 15^{th}, 40^{th}, 70^{th} and 110^{th} iteration (left) or the 15^{th}, 40^{th}, 90^{th}, 160^{th} iteration (right). These selected iterations correspond to a relative misfit decrease of approximately 30%, 10%, 3% and 1% respectively.

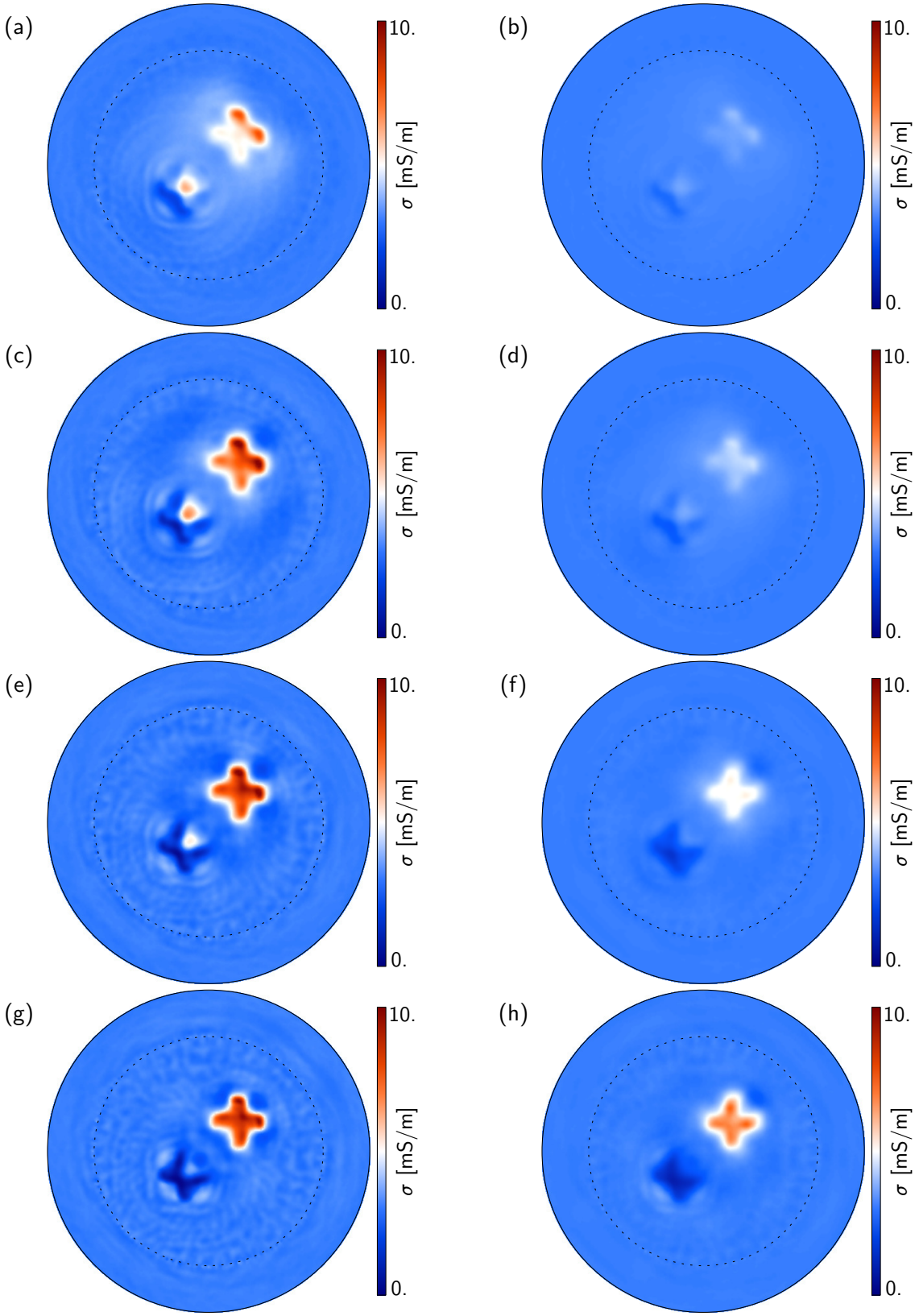


Figure D.2: Intermediate inversion results (conductivity) for the line search *l*-BFGS algorithm with two different scaling for the inner product: $\beta = 2$ (left) and $\beta = 5$ (right). From top to bottom, these reconstruction corresponds to the 15^{th}, 40^{th}, 70^{th} and 110^{th} iteration (left) or the 15^{th}, 40^{th}, 90^{th}, 160^{th} iteration (right). These selected iterations correspond to a relative misfit decrease of approximately 30%, 10%, 3% and 1% respectively.

Appendix E

Total variation penalization

The total variation penalization term[1] is given by

$$J_m(m) \triangleq \int_\Omega |\nabla m| \; d\Omega. \tag{E.1}$$

Its series development in the perturbed direction δm is therefore successively given by

$$\begin{aligned}
J_m(m+\delta m) &= \int_\Omega |\nabla (m+\delta m)| \; d\Omega && \text{(E.2)}\\
&= \int_\Omega \sqrt{\overline{(\nabla (m+\delta m))} (\nabla (m+\delta m))} \, d\Omega && \text{(E.3)}\\
&\approx \int_\Omega \sqrt{\overline{\nabla m} \nabla m} \, d\Omega && \text{(E.4)}\\
&\qquad + \int_\Omega \frac{1}{2} \frac{1}{\sqrt{\overline{\nabla m} \nabla m}} \left(\overline{\nabla m} \nabla \delta m + \overline{\nabla \delta m} \nabla m + \overline{\nabla \delta m} \nabla \delta m \right) \, d\Omega && \text{(E.5)}\\
&\approx J_m(m) + \frac{1}{2} \int_\Omega \frac{1}{|\nabla m|} 2\mathrm{Re} \left(\overline{\nabla m} \nabla \delta m \right) \, d\Omega && \text{(E.6)}\\
&\approx J_m(m) + \mathrm{Re} \int_\Omega \frac{\overline{\nabla m}}{|\nabla m|} \nabla \delta m \, d\Omega && \text{(E.7)}
\end{aligned}$$

such that its first order directional derivatives is given by

$$\{D_m J_m(m)\}(\delta m) = \mathrm{Re} \int_\Omega \frac{\overline{\nabla m}}{|\nabla m|} \nabla \delta m \, d\Omega. \tag{E.8}$$

Similarly, a series development of the directional derivative in the direction δm_1, perturbed in a second direction δm_2 successively gives

$$\begin{aligned}
&\{D_m J_m(m+\delta m_2)\}(\delta m_1) && \text{(E.9)}\\
&\quad = \mathrm{Re} \int_\Omega \frac{\overline{\nabla (m+\delta m_2)}}{|\nabla (m+\delta m_2)|} \nabla \delta m_1 \, d\Omega && \text{(E.10)}
\end{aligned}$$

[1] For compactness, the spatial gradient **grad** $(\cdot)$ is denoted ∇ throughout this appendix.

$$\approx \mathrm{Re}\int_\Omega \overline{\nabla\,(m+\delta m_2)}\nabla\delta m_1\left[\frac{1}{|\nabla m|}-\frac{1}{2}\frac{1}{|\nabla m|^3}2\mathrm{Re}\left(\overline{\nabla\delta m_2}\nabla m\right)\right]d\Omega \tag{E.11}$$

$$\approx \mathrm{Re}\int_\Omega \left[\frac{\overline{\nabla\,(m+\delta m_2)}}{|\nabla m|}\nabla\delta m_1-\overline{\nabla m}\nabla\delta m_1\frac{1}{|\nabla m|^3}\mathrm{Re}\left(\overline{\nabla\delta m_2}\nabla m\right)\right]d\Omega \tag{E.12}$$

$$\approx \mathrm{Re}\int_\Omega \left[\frac{\overline{\nabla m}}{|\nabla m|}\nabla\delta m_1+\frac{\overline{\nabla\delta m_2}}{|\nabla m|}\nabla\delta m_1-\frac{\overline{\nabla m}\nabla\delta m_1}{|\nabla m|^3}\mathrm{Re}\left(\overline{\nabla\delta m_2}\nabla m\right)\right]d\Omega \tag{E.13}$$

$$\approx \{D_m J_m(m)\}(\delta m_1) \tag{E.14}$$

$$+\,\mathrm{Re}\int_\Omega \frac{1}{|\nabla m|}\left[\overline{\nabla\delta m_2}\nabla\delta m_1-\mathrm{Re}\left(\nabla\delta m_1\frac{\overline{\nabla m}}{|\nabla m|}\right)\mathrm{Re}\left(\frac{\nabla m}{|\nabla m|}\overline{\nabla\delta m_2}\right)\right]d\Omega \tag{E.15}$$

such that the second order directional derivative is given by

$$\{D^2_{mm}J_m(m)\}(\delta m_1)(\delta m_2) = \mathrm{Re}\int_\Omega \frac{1}{|\nabla m|}\left[\overline{\nabla\delta m_2}\nabla\delta m_1-\mathrm{Re}\left(\overline{\nabla\delta m_2}\frac{\nabla m}{|\nabla m|}\right)\mathrm{Re}\left(\frac{\overline{\nabla m}}{|\nabla m|}\nabla\delta m_1\right)\right]d\Omega. \tag{E.16}$$

For real-valued model parameters, it further simplifies as

$$\{D^2_{mm}J_m(m)\}(\delta m_1)(\delta m_2) = \int_\Omega \frac{1}{|\nabla m|}\left[\nabla\delta m_2\nabla\delta m_1-\nabla\delta m_2\frac{\nabla m\nabla m}{|\nabla m|^2}\nabla\delta m_1\right]d\Omega. \tag{E.17}$$

which can be written compactly as

$$\{D^2_{mm}J_m(m)\}(\delta m_1)(\delta m_2) = \int_\Omega g_\perp(|\nabla m|^2)\;\nabla\delta m_2\,\boldsymbol{D}_\perp\,\nabla\delta m_1\,d\Omega \tag{E.18}$$

with

$$g_\perp(r^2)=\frac{1}{r},\qquad \boldsymbol{D}_\perp=\boldsymbol{I}-\boldsymbol{D}_\parallel\quad\text{and}\quad \boldsymbol{D}_\parallel:=\frac{\nabla m\nabla m}{|\nabla m|^2}. \tag{E.19}$$

Bibliography

[1] X. Adriaens, F. Henrotte, and C. Geuzaine. "Adjoint state method for time-harmonic scattering problems with boundary perturbations". In: *J. Comput. Phys.* 428 (Mar. 2021), p. 109981.

[2] X. Adriaens, L. Métivier, and C. Geuzaine. "A trust-region Newton method for frequency-domain full-waveform inversion". In: *First Int. Meet. Appl. Geosci. Energy Expand. Abstr.* Vol. 2021-Septe. Society of Exploration Geophysicists, Sept. 2021, pp. 757–761.

[3] X. Adriaens, L. Métivier, and C. Geuzaine. "Inner product preconditioned trust-region methods for frequency-domain full waveform inversion". In: *Review* ().

[4] H. S. Aghamiry, A. Gholami, and S. Operto. "Full waveform inversion by proximal Newton method using adaptive regularization". In: *Geophys. J. Int.* 224.1 (2021), pp. 169–180.

[5] V. Akçelik, K. Ko, L. Q. Lee, et al. "Shape determination for deformed electromagnetic cavities". In: *J. Comput. Phys.* 227.3 (2008), pp. 1722–1738.

[6] M. Alalade, I. Reichert, D. Köhn, F. Wuttke, and T. Lahmer. "A Cyclic Multi-Stage Implementation of the Full-Waveform Inversion for the Identification of Anomalies in Dams". In: *Infrastructures* 7.12 (Nov. 2022), p. 161.

[7] T. Alkhalifah and R.-É. Plessix. "A recipe for practical full-waveform inversion in anisotropic media: An analytical parameter resolution study". In: *GEOPHYSICS* 79.3 (May 2014), R91–R101.

[8] A. Y. Anagaw and M. D. Sacchi. "Edge-preserving seismic imaging using the total variation method". In: *J. Geophys. Eng.* 9.2 (2012), pp. 138–146.

[9] A. Y. Anagaw and M. D. Sacchi. "Model parametrization strategies for Newton-based acoustic full waveform inversion". In: *J. Appl. Geophys.* 157 (2018), pp. 23–36.

[10] A. H. Andersen. "A Ray Tracing Approach to Restoration and Resolution Enhancement in Experimental Ultrasound Tomography". In: *Ultrason. Imaging* 12.4 (1990), pp. 268–291.

[11] A. Asnaashari, R. Brossier, S. Garambois, et al. "Regularized seismic full waveform inversion with prior model information". In: *GEOPHYSICS* 78.2 (Mar. 2013), R25–R36.

[12] P. M. van den Berg, A. Abubakar, and J. T. Fokkema. "Multiplicative regularization for contrast profile inversion". In: *Radio Sci.* 38.2 (Apr. 2003).

[13] S. Bernard, V. Monteiller, D. Komatitsch, and P. Lasaygues. "Ultrasonic computed tomography based on full-waveform inversion for bone quantitative imaging". In: *Phys. Med. Biol.* 62.17 (Aug. 2017), pp. 7011–7035.

[14] S. Biton, N. Arbel, G. Drozdov, G. Gilboa, and A. Rosenthal. "Optoacoustic model-based inversion using anisotropic adaptive total-variation regularization". In: *Photoacoustics* 16.July 2018 (Dec. 2019), p. 100142.

[15] M. Bonazzoli, H. Haddar, and T. A. Vu. "Convergence analysis of multi-step one-shot methods for linear inverse problems". In: *arXiv Prepr.* (July 2022).

[16] P. Bouboulis and S. Theodoridis. "Extension of wirtinger's calculus to reproducing kernel hilbert spaces and the complex kernel LMS". In: *IEEE Trans. Signal Process.* 59.3 (2011), pp. 964–978.

[17] R. Brossier. "Two-dimensional frequency-domain visco-elastic full waveform inversion: Parallel algorithms, optimization and performance". In: *Comput. Geosci.* 37.4 (Apr. 2011), pp. 444–455.

[18] R. Brossier, S. Operto, and J. Virieux. "Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion". In: *GEOPHYSICS* 74.6 (Nov. 2009), WCC105–WCC118.

[19] R. Brossier, S. Operto, and J. Virieux. "Which data residual norm for robust elastic frequency-domain full waveform inversion?" In: *GEOPHYSICS* 75.3 (May 2010), R37–R46.

[20] T. Brox, R. van den Boomgaard, F. Lauze, et al. "Adaptive Structure Tensors and their Applications". In: 2006. Chap. 2, pp. 17–47.

[21] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler. "A Computational Framework for Infinite-Dimensional Bayesian Inverse Problems Part I: The Linearized Case, with Application to Global Seismic Inversion". In: *SIAM J. Sci. Comput.* 35.6 (Jan. 2013), A2494–A2523.

[22] C. Bunks, F. M. Saleck, S. Zaleski, and G. Chavent. "Multiscale seismic waveform inversion". In: *GEOPHYSICS* 60.5 (Sept. 1995), pp. 1457–1473.

[23] F. Cakoni, D. Colton, and P. Monk. "The determination of boundary coefficients from far field measurements". In: *J. Integr. Equations Appl.* 22.2 (2010), pp. 167–191.

[24] Y. Capdeville and L. Métivier. "Elastic full waveform inversion based on the homogenization method: Theoretical framework and 2-D numerical illustrations". In: *Geophys. J. Int.* 213.2 (2018), pp. 1093–1112.

[25] M. d. S. R. Carneiro, B. Pereira-Dias, D. M. Soares Filho, and L. Landau. "On the Scaling of the Update Direction for Multi-parameter Full Waveform Inversion: Applications to 2D Acoustic and Elastic Cases". In: *Pure Appl. Geophys.* 175.1 (Jan. 2018), pp. 217–241.

[26] E. Causse, R. Mittet, and B. Ursin. "Preconditioning of full-waveform inversion in viscoacoustic media". In: *GEOPHYSICS* 64.1 (Jan. 1999), pp. 130–145.

[27] T. Chan, A. Marquina, and P. Mulet. "High-Order Total Variation-Based Image Restoration". In: *SIAM J. Sci. Comput.* 22.2 (Jan. 2000), pp. 503–516.

[28] G. Chavent. "Identification of Distributed Parameter Systems: About the Output Least Square Method, its Implementation, and Identifiability". In: *IFAC Proc. Vol.* 12.8 (Sept. 1979), pp. 85–97.

[29] X. Chen. *Computational Methods for Electromagnetic Inverse Scattering*. Singapore: John Wiley & Sons Singapore Pte. Ltd, May 2018.

[30] B. Chi, L. Dong, and Y. Liu. "Full waveform inversion method using envelope objective function without low frequency data". In: *J. Appl. Geophys.* 109 (2014), pp. 36–46.

[31] Y. Choi and T. Alkhalifah. "Application of multi-source waveform inversion to marine streamer data using the global correlation norm". In: *Geophys. Prospect.* 60.4 (July 2012), pp. 748–758.

[32] Y. Choi and T. Alkhalifah. "Unwrapped phase inversion with an exponential damping". In: *GEOPHYSICS* 80.5 (2015), R251–R264.

[33] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Vol. 93. Applied Mathematical Sciences. Cham: Springer International Publishing, 2019.

[34] B. Consolvo, M. Zuberi, R. Pratt, and P. Cary. "FWI with Scaled-Sobolev Preconditioning Applied to Short-offset Vibroseis Field Data". In: *79th EAGE Conf. Exhib. 2017*. June 2017, pp. 10–15.

[35] J. Cudeiro-Blanco, C. Cueto, O. Bates, et al. "Design and Construction of a Low-Frequency Ultrasound Acquisition Device for 2-D Brain Imaging Using Full-Waveform Inversion". In: *Ultrasound Med. Biol.* 48.10 (Oct. 2022), pp. 1995–2008.

[36] D. Datta and M. K. Sen. "Estimating a starting model for full-waveform inversion using a global optimization method". In: *GEOPHYSICS* 81.4 (July 2016), R211–R223.

[37] H. A. Debens, M. Warner, A. Umpleby, and N. V. Da Silva. "Global anisotropic 3D FWI". In: *SEG Tech. Progr. Expand. Abstr.* 34 (2015), pp. 1193–1197.

[38] L. Demanet, P.-D. Létourneau, N. Boumal, et al. "Matrix probing: A randomized preconditioner for the wave-equation Hessian". In: *Appl. Comput. Harmon. Anal.* 32.2 (Mar. 2012), pp. 155–168.

[39] A. J. Devaney and M. L. Oristaglio. "Geophysical diffraction tomography". In: *1984 SEG Annu. Meet. SEG 1984* 1 (1984), pp. 330–333.

[40] A. J. Devaney. *Mathematical foundations of imaging, tomography and wavefield inversion*. 2012, pp. 1–518.

[41] S. Didas, J. Weickert, and B. Burgeth. "Properties of Higher Order Nonlinear Diffusion Filtering". In: *J. Math. Imaging Vis.* 35.3 (Nov. 2009), pp. 208–226.

[42] O. Dorn, H. Bertete-Aguirre, and G. C. Papanicolaou. "Adjoint fields and sensitivities for 3D electromagnetic imaging in isotropic and anisotropic media". In: *Lect. Notes Math.* 1943 (2008), pp. 35–65.

[43] S. C. Eisenstat and H. F. Walker. "Choosing the forcing terms in an inexact Newton method". In: *SIAM J. Sci. Comput.* 17.1 (1996), pp. 16–32.

[44] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Dordrecht: Springer Netherlands, 1996.

[45] I. Epanomeritakis, V. Akçelik, O. Ghattas, and J. Bielak. "A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion". In: *Inverse Probl.* 24.3 (June 2008), p. 034015.

[46] J. Ernst, H. Maurer, A. Green, and K. Holliger. "Full-Waveform Inversion of Crosshole Radar Data Based on 2-D Finite-Difference Time-Domain Solutions of Maxwell's Equations". In: *IEEE Trans. Geosci. Remote Sens.* 45.9 (Sept. 2007), pp. 2807–2828.

[47] J. R. Ernst, A. G. Green, H. Maurer, and K. Holliger. "Application of a new 2D time-domain full-waveform inversion scheme to crosshole radar data". In: *GEOPHYSICS* 72.5 (Sept. 2007), J53–J64.

[48] O. G. Ernst and M. J. Gander. "Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods". In: *Numer. Anal. multiscale Probl.* 83 (2012), pp. 325–363.

[49] E. Esser, L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann. "Total Variation Regularization Strategies in Full-Waveform Inversion". In: *SIAM J. Imaging Sci.* 11.1 (2018), pp. 376–406.

[50] J.-y. Fan and Y.-x. Yuan. "A new trust region algorithm with trust region radius converging to zero". In: *Proc. 5th Int. Conf. Optim. Tech. Appl.* February (2001).

[51] J. Fan, J. Pan, and H. Song. "A Retrospective Trust Region Algorithm with Trust Region Converging to Zero". In: *J. Comput. Math.* 34.4 (June 2016), pp. 421–436.

[52] Y. Favennec, F. Dubot, D. Le Hardy, B. Rousseau, and D. R. Rousse. "Space-Dependent Sobolev Gradients as a Regularization for Inverse Radiative Transfer Problems". In: *Math. Probl. Eng.* 2016 (2016), pp. 1–18.

[53] A. Fichtner and T. van Leeuwen. "Resolution analysis by random probing". In: *J. Geophys. Res. Solid Earth* 120.8 (Aug. 2015), pp. 5549–5573.

[54] A. Fichtner and J. Trampert. "Hessian kernels of seismic data functionals based upon adjoint techniques". In: *Geophys. J. Int.* 185.2 (May 2011), pp. 775–798.

[55] A. Fichtner and J. Trampert. "Resolution analysis in full waveform inversion". In: *Geophys. J. Int.* 187.3 (2011), pp. 1604–1624.

[56] A. Fichtner, J. Trampert, P. Cupillard, et al. "Multiscale full waveform inversion". In: *Geophys. J. Int.* 194.1 (2013), pp. 534–556.

[57] A. Fichtner, A. Zunino, and L. Gebraad. "Hamiltonian Monte Carlo solution of tomographic inverse problems". In: *Geophys. J. Int.* 216.2 (Feb. 2019), pp. 1344–1363.

[58] S. Fomel. "Shaping regularization in geophysical-estimation problems". In: *GEOPHYSICS* 72.2 (Mar. 2007), R29–R36.

[59] K. Gao and L. Huang. "Acoustic- and elastic-waveform inversion with total generalized p-variation regularization". In: *Geophys. J. Int.* 218.2 (2019), pp. 933–957.

[60] N. Georgieva. "Feasible adjoint sensitivity technique for EM design optimization". In: *IEEE Trans. Microw. Theory Tech.* 2 (2002), pp. 971–974.

[61] C. Geuzaine and J.-F. Remacle. "Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities". In: *Int. J. Numer. Methods Eng. 79(11)* 0 (2009), pp. 1309–1331.

[62] M. Ghassemi, N. Sangary, and M. Bakr. "Antenna design exploiting adjoint sensitivity-based geometry evolution". In: *IET Microwaves, Antennas Propag.* 7.4 (2013), pp. 268–276.

[63] M. Giles and N. Pierce. "Adjoint equations in CFD - Duality, boundary conditions and solution behaviour". In: *13th Comput. Fluid Dyn. Conf.* Reston, Virigina: American Institute of Aeronautics and Astronautics, June 1997.

[64] A. Gokhberg and A. Fichtner. "Full-waveform inversion on heterogeneous HPC systems". In: *Comput. Geosci.* 89 (Apr. 2016), pp. 260–268.

[65] M. Grasmair and F. Lenzen. "Anisotropic total variation filtering". In: *Appl. Math. Optim.* 62.3 (2010), pp. 323–339.

[66] M. J. Grote, M. Kray, and U. Nahum. "Adaptive eigenspace method for inverse scattering problems in the frequency domain". In: *Inverse Probl.* 33.2 (Feb. 2017), p. 025006.

[67] M. J. Grote and U. Nahum. "Adaptive eigenspace for multi-parameter inverse scattering problems". In: *Comput. Math. with Appl.* 77.12 (June 2019), pp. 3264–3280.

[68] L. Guasch, O. Calderón Agudo, M.-X. Tang, P. Nachev, and M. Warner. "Full-waveform inversion imaging of the human brain". In: *npj Digit. Med.* 3.1 (Dec. 2020), p. 28.

[69] A. Guitton. "Blocky regularization schemes for Full-Waveform Inversion". In: *Geophys. Prospect.* 60.5 (2012), pp. 870–884.

[70] A. Guitton, G. Ayeni, and E. Díaz. "Constrained full-waveform inversion by model reparameterization". In: *GEOPHYSICS* 77.2 (Mar. 2012), R117–R127.

[71] R. Haftka. "Structural Optimization System". In: *J. Acoust. Soc. Am.* 129.3 (2011), p. 1666.

[72] B. Hak and W. A. Mulder. "Migration for velocity and attenuation perturbations". In: *Geophys. Prospect.* 58.6 (Feb. 2010), pp. 939–952.

[73] D. Hale. "Local dip filtering with directional Laplacians". In: *CWP Rep.* (2007), pp. 91–102.

[74] P. Hansen and L. Hesselink. "Accurate adjoint design sensitivities for nano metal optics". In: *Opt. Express* 23.18 (Sept. 2015), p. 23899.

[75] P. C. Hansen and D. P. O'Leary. "The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems". In: *SIAM J. Sci. Comput.* 14.6 (Nov. 1993), pp. 1487–1503.

[76] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Vol. 23. Mathematical Modelling: Theory and Applications. Dordrecht: Spinger, 2009.

[77] R. Hiptmair and J. Li. "Shape derivatives for scattering problems". In: *Inverse Probl.* 34.10 (Oct. 2018), p. 105001.

[78] W. Hu, A. Abubakar, T. M. Habashy, and J. Liu. "Preconditioned non-linear conjugate gradient method for frequency domain full-waveform seismic inversion". In: *Geophys. Prospect.* 59.3 (May 2011), pp. 477–491.

[79] W. Hu, A. Abubakar, and T. M. Habashy. "Simultaneous multifrequency inversion of full-waveform seismic data". In: *GEOPHYSICS* 74.2 (Mar. 2009), R1–R14.

[80] J. S. Jensen and O. Sigmund. "Topology optimization for nano-photonics". In: *Laser Photonics Rev.* 5.2 (2011), pp. 308–321.

[81] P. Jolivet and P.-H. Tournier. "Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers". In: *SC16 Int. Conf. High Perform. Comput. Networking, Storage Anal.* Vol. 0. IEEE, Nov. 2016, pp. 190–203.

[82] S. I. Kabanikhin. "Definitions and examples of inverse and ill-posed problems". In: *J. Inverse Ill-posed Probl.* 16.4 (Jan. 2008), pp. 317–357.

[83] A. Kadu, T. van Leeuwen, and W. A. Mulder. "Salt Reconstruction in Full-Waveform Inversion With a Parametric Level-Set Method". In: *IEEE Trans. Comput. Imaging* 3.2 (June 2017), pp. 305–315.

[84] M. Kalita, V. Kazei, Y. Choi, and T. Alkhalifah. "Regularized full-waveform inversion with automated salt flooding". In: *GEOPHYSICS* 84.4 (July 2019), R569–R582.

[85] A. Kalogeropoulos, J. van der Kruk, J. Hugenschmidt, S. Busch, and K. Merz. "Chlorides and moisture assessment in concrete by GPR full waveform inversion". In: *Near Surf. Geophys.* 9.3 (June 2011), pp. 277–286.

[86] B. Kaltenbacher, A. Neubauer, and O. Scherzer. *Iterative Regularization Methods for Nonlinear Ill-Posed Problems*. De Gruyter, May 2008.

[87] R. Kamei and R. G. Pratt. "Inversion strategies for visco-acoustic waveform inversion". In: *Geophys. J. Int.* 194.2 (Aug. 2013), pp. 859–884.

[88] P. Kazemi and R. Renka. "Minimization of the Ginzburg–Landau energy functional by a Sobolev gradient trust-region method". In: *Appl. Math. Comput.* 219.11 (Feb. 2013), pp. 5936–5942.

[89] A. Klotzsche, J. van der Kruk, G. Angelo Meles, et al. "Full-waveform inversion of cross-hole ground-penetrating radar data to characterize a gravel aquifer close to the Thur River, Switzerland". In: *Near Surf. Geophys.* 8.6 (Dec. 2010), pp. 635–649.

[90] A. Klotzsche, H. Vereecken, and J. Van Der Kruk. "Review of crosshole ground-penetrating radar full-waveform inversion of experimental data: Recent developments, challenges, and pitfalls". In: *Geophysics* 84.6 (2019), H13–H28.

[91] K. Koolstra and R. Remis. "Learning a preconditioner to accelerate compressed sensing reconstructions in MRI". In: *Magn. Reson. Med.* 87.4 (Apr. 2022), pp. 2063–2073.

[92] A. Kordjazi, J. T. Coe, and M. Afanasiev. "Nondestructive Evaluation of Drilled Shaft Construction Anomalies Using Full Waveform Tomography of Simulated Crosshole Measurements". In: *J. Nondestruct. Eval.* 40.1 (Mar. 2021), p. 3.

[93] R. Kress and W. Rundell. "Inverse scattering for shape and impedance". In: *Inverse Probl.* 17.4 (Aug. 2001), pp. 1075–1085.

[94] K. Kreutz-Delgado. "The Complex Gradient Operator and the CR-Calculus". In: (June 2009).

[95] T. Lahmer, M. Schickert, and I. Reichert. "Possibilities to Enhance Tomography Imaging of Concrete Structures by the Full Waveform Inversion". In: *5th Int. Conf. Smart Monit. Assess. Rehabil. Civ. Struct.* 2019, pp. 1–6.

[96] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch. "Adjoint shape optimization applied to electromagnetic design". In: *Opt. Express* 21.18 (2013), p. 21693.

[97] F. Lavoué, R. Brossier, L. Métivier, S. Garambois, and J. Virieux. "Two-dimensional permittivity and conductivity imaging by full waveform inversion of multioffset GPR data: a frequency-domain quasi-Newton approach". In: *Geophys. J. Int.* 197.1 (Apr. 2014), pp. 248–268.

[98] T. van Leeuwen and F. J. Herrmann. "Mitigating local minima in full-waveform inversion by expanding the search space". In: *Geophys. J. Int.* 195.1 (2013), pp. 661–667.

[99] W. Lewis, D. Amazonas, D. Vigh, and R. Coates. "Geologically constrained full-waveform inversion using an anisotropic diffusion based regularization scheme: Application to a 3D offshore Brazil dataset". In: *SEG Tech. Progr. Expand. Abstr. 2014*. November. Society of Exploration Geophysicists, Aug. 2014, pp. 1083–1088.

[100] D. Li, F. Gao, and P. Williamson. "A deep learning approach for acoustic FWI with elastic data". In: *SEG Tech. Progr. Expand. Abstr. 2019*. Society of Exploration Geophysicists, Aug. 2019, pp. 2303–2307.

[101] Y. E. Li and L. Demanet. "Full-waveform inversion with extrapolated low-frequency data". In: *GEOPHYSICS* 81.6 (Nov. 2016), R339–R348.

[102] P. Lin, S. Peng, Y. Lu, and W. Du. "The trust region method for time-domain full waveform inversion". In: *Proc. 7th Int. Conf. Environ. Eng. Geophys. Summit Forum Chinese Acad. Eng. Eng. Sci. Technol.* Paris, France: Atlantis Press, 2016, pp. 220–223.

[103] Y. Lin and L. Huang. "Acoustic-and elastic-waveform inversion using a modified total-variation regularization scheme". In: *Geophys. J. Int.* 200.1 (2014), pp. 489–502.

[104] J. J. Liu, G. Nakamura, and M. Sini. "Reconstruction of the shape and surface impedance from acoustic scattering data for an arbitrary cylinder". In: *SIAM J. Appl. Math.* 67.4 (2007), pp. 1124–1146.

[105] Q. Liu, D. Peter, and C. Tape. "Square-root variable metric based elastic full-waveform inversion – Part 1: theory and validation". In: *Geophys. J. Int.* 218.2 (Aug. 2019), pp. 1121–1135.

[106] Q. Liu, D. Peter, and C. Tape. "Square-root variable metric based elastic full-waveform inversion – Part 1: theory and validation". In: *Geophys. J. Int.* 218.2 (Aug. 2019), pp. 1121–1135.

[107] I. Loris, H. Douma, G. Nolet, I. Daubechies, and C. Regone. "Nonlinear regularization techniques for seismic tomography". In: *J. Comput. Phys.* 229.3 (2010), pp. 890–905.

[108] F. Lucka, M. Pérez-Liva, B. E. Treeby, and B. T. Cox. "High resolution 3D ultrasonic breast imaging by time-domain full waveform inversion". In: *Inverse Probl.* 38.2 (Feb. 2022), p. 025008.

[109] J. Lysmer and R. L. Kuhlemeyer. "Finite dynamic model for infinite media". In: *J. Eng. Mech. Div.* 95.4 (1969), pp. 859–878.

[110] G. S. Martin, R. Wiley, and K. J. Marfurt. "Marmousi2: An elastic upgrade for Marmousi". In: *Lead. Edge* 25.2 (Feb. 2006), pp. 156–166.

[111] G. A. Meles, S. A. Greenhalgh, A. G. Green, H. Maurer, and J. Van der Kruk. "GPR Full-Waveform Sensitivity and Resolution Analysis Using an FDTD Adjoint Method". In: *IEEE Trans. Geosci. Remote Sens.* 50.5 (May 2012), pp. 1881–1896.

[112] G. A. Meles, J. Van der Kruk, S. A. Greenhalgh, et al. "A New Vector Waveform Inversion Algorithm for Simultaneous Updating of Conductivity and Permittivity Parameters From Combination Crosshole/Borehole-to-Surface GPR Data". In: *IEEE Trans. Geosci. Remote Sens.* 48.9 (Sept. 2010), pp. 3391–3407.

[113] L. Métivier, F. Bretaudeau, R. Brossier, S. Operto, and J. Virieux. "Full waveform inversion and the truncated Newton method: quantitative imaging of complex subsurface structures". In: *Geophys. Prospect.* 62.6 (Nov. 2014), pp. 1353–1375.

[114] L. Métivier and R. Brossier. "On the use of nonlinear anisotropic diffusion filters for seismic imaging using the full waveform". In: *Inverse Probl.* 38.11 (Nov. 2022), p. 115001.

[115] L. Métivier, R. Brossier, Q. Mérigot, E. Oudet, and J. Virieux. "Measuring the misfit between seismograms using an optimal transport distance: application to full waveform inversion". In: *Geophys. J. Int.* 205.1 (Apr. 2016), pp. 345–377.

[116] L. Métivier, R. Brossier, S. Operto, and J. Virieux. "Acoustic multi-parameter FWI for the reconstruction of P-wave velocity, density and attenuation: preconditioned truncated Newton approach". In: *SEG Tech. Progr. Expand. Abstr. 2015*. Vol. 34. Society of Exploration Geophysicists, Aug. 2015, pp. 1198–1203.

[117] L. Métivier, R. Brossier, S. Operto, and J. Virieux. "Full Waveform Inversion and the Truncated Newton Method". In: *SIAM Rev.* 59.1 (Jan. 2017), pp. 153–195.

[118] L. Métivier, R. Brossier, J. Virieux, and S. Operto. "Full Waveform Inversion and the Truncated Newton Method". In: *SIAM J. Sci. Comput.* 35.2 (Jan. 2013), B401–B437.

[119] M. Meyer, J. P. Hermand, M. Asch, and J. C. Le Gac. "An analytic multiple frequency adjoint-based inversion algorithm for parabolic-type approximations in ocean acoustics". In: *Inverse Probl. Sci. Eng.* 14.3 (2006), pp. 245–265.

[120] J. Minet, P. Bogaert, M. Vanclooster, and S. Lambot. "Validation of ground penetrating radar full-waveform inversion for field scale soil moisture mapping". In: *J. Hydrol.* 424-425 (Mar. 2012), pp. 112–123.

[121] P. Mojabi and J. LoVetri. "Use of Wirtinger calculus in Gauss-Newton inversion of microwave tomography data". In: *2014 XXXIth URSI Gen. Assem. Sci. Symp. (URSI GASS)*. 2. IEEE, Aug. 2014, pp. 1–4.

[122] P. Mora. "Nonlinear two-dimensional elastic inversion of multioffset seismic data". In: *GEOPHYSICS* 52.9 (Sept. 1987), pp. 1211–1228.

[123] L. Mosser, O. Dubrule, and M. J. Blunt. "Stochastic Seismic Waveform Inversion Using Generative Adversarial Networks as a Geological Prior". In: *Math. Geosci.* 52.1 (2020), pp. 53–79.

[124] W. Mulder and R.-É. Plessix. "Exploring some issues in acoustic full waveform inversion". In: *Geophys. Prospect.* 56.6 (Nov. 2008), pp. 827–841.

[125] F. Natterer. *The Mathematics of Computerized Tomography*. Wiesbaden: Vieweg+Teubner Verlag, 1986.

[126] A. G. Nekut. "Electromagnetic ray-trace tomography". In: *GEOPHYSICS* 59.3 (Mar. 1994), pp. 371–377.

[127] J. Neuberger. *Sobolev Gradients and Differential Equations*. Vol. 1670. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[128] N. M. Newmark. "A Method of Computation for Structural Dynamics". In: *J. Eng. Mech. Div.* 85.3 (July 1959), pp. 67–94.

[129] A. C. R. Niederberger, D. A. Fattal, N. R. Gauger, S. Fan, and R. G. Beausoleil. "Sensitivity analysis and optimization of sub-wavelength optical gratings using adjoints". In: *Opt. Express* 22.11 (2014), p. 12971.

[130] J. Nocedal, S. J. Wright, and S. M. Robinson. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006, pp. 1–651.

[131] Y. Noguchi and T. Yamada. "Topology optimization of acoustic metasurfaces by using a two-scale homogenization method". In: *Appl. Math. Model.* 98 (2021), pp. 465–497.

[132] J.-W. Oh, M. Kalita, and T. Alkhalifah. "3D elastic full-waveform inversion using P-wave excitation amplitude: Application to ocean bottom cable field data". In: *GEOPHYSICS* 83.2 (Mar. 2018), R129–R140.

[133] S. Operto, Y. Gholami, V. Prieux, et al. "A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice". In: *Lead. Edge* 32.9 (Sept. 2013), pp. 1040–1054.

[134] W. Pan, K. A. Innanen, G. F. Margrave, et al. "Estimation of elastic constants for HTI media using Gauss-Newton and full-Newton multiparameter full-waveform inversion". In: *GEOPHYSICS* 81.5 (Sept. 2016), R275–R291.

[135] W. Pan, K. A. Innanen, and W. Liao. "Accelerating Hessian-free Gauss-Newton full-waveform inversion via l-BFGS preconditioned conjugate-gradient algorithm". In: *GEOPHYSICS* 82.2 (Mar. 2017), R49–R64.

[136] W. Pan, K. A. Innanen, and W. Liao. "Accelerating Hessian-free Gauss-Newton full-waveform inversion via improved preconditioning strategies". In: *SEG Tech. Progr. Expand. Abstr. 2016*. Vol. 35. Society of Exploration Geophysicists, Sept. 2016, pp. 1455–1461.

[137] O. Pankratov and A. Kuvshinov. "General formalism for the efficient calculation of the Hessian matrix of EM data misfit and Hessian-vector products based upon adjoint sources approach". In: *Geophys. J. Int.* 200.3 (Mar. 2015), pp. 1449–1465.

[138] B. Park, W. Ha, and C. Shin. "A comparison of the preconditioning effects of different parameterization methods for monoparameter full waveform inversions in the Laplace domain". In: *J. Appl. Geophys.* 172 (2020), p. 103883.

[139] C. Patriarca, S. Lambot, M. Mahmoudzadeh, J. Minet, and E. Slob. "Reconstruction of sub-wavelength fractures and physical properties of masonry media using full-waveform inversion of proximal penetrating radar". In: *J. Appl. Geophys.* 74.1 (May 2011), pp. 26–37.

[140] B. Peters and F. J. Herrmann. "Constraints versus penalties for edge-preserving full-waveform inversion". In: *Lead. Edge* 36.1 (Jan. 2017), pp. 94–100.

[141] E. R. Pike and P. C. Sabatier. *Scattering, Two-Volume Set: Scattering and Inverse Scattering in Pure and Applied Science*. Elsevier, 2001.

[142] R.-É. Plessix. "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications". In: *Geophys. J. Int.* 167.2 (Nov. 2006), pp. 495–503.

[143] R.-É. Plessix and Q. Cao. "A parametrization study for surface seismic full waveform inversion in an acoustic vertical transversely isotropic medium". In: *Geophys. J. Int.* 185.1 (2011), pp. 539–556.

[144] R.-É. Plessix and W. A. Mulder. "Frequency-domain finite-difference amplitude-preserving migration". In: *Geophys. J. Int.* 157.3 (June 2004), pp. 975–987.

[145] R.-É. Plessix and W. A. Mulder. "Resistivity imaging with controlled-source electromagnetic data: depth and data weighting". In: *Inverse Probl.* 24.3 (June 2008), pp. 1–22.

[146] R. G. Pratt and M. H. Worthington. "Inverse theory applied to multi-source cross-hole tomography. Part 1: acoustic wave-equation method". In: *Int. J. Rock Mech. Min. Sci. Geomech. Abstr.* 27.6 (Dec. 1990), p. 355.

[147] R. G. Pratt, L. Huang, N. Duric, and P. Littrup. "Sound-speed and attenuation imaging of breast tissue using waveform tomography of transmission ultrasound data". In: *Med. Imaging 2007 Phys. Med. Imaging*. Ed. by J. Hsieh and M. J. Flynn. Vol. 6510. Mar. 2007, 65104S.

[148] R. G. Pratt, C. Shin, and G. J. Hicks. "Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion". In: *Geophys. J. Int.* 133.2 (1998), pp. 341–362.

[149] V. Prieux, R. Brossier, S. Operto, and J. Virieux. "Multiparameter full waveform inversion of multicomponent ocean-bottom-cable data from the Valhall field. Part 1: imaging compressional wave speed, density and attenuation". In: *Geophys. J. Int.* 194.3 (Sept. 2013), pp. 1640–1664.

[150] J. Primus, E. Piot, and F. Simon. "An adjoint-based method for liner impedance eduction: Validation and numerical investigation". In: *J. Sound Vib.* 332.1 (Jan. 2013), pp. 58–75.

[151] J. Primus, E. Piot, and F. Simon. "An adjoint-based method for liner impedance eduction: Validation and numerical investigation". In: *J. Sound Vib.* 332.1 (2013), pp. 58–75.

[152] A. C. Ramírez and W. R. Lewis. "Regularization and full-waveform inversion: A two-step approach". In: *SEG Tech. Progr. Expand. Abstr. 2010*. Society of Exploration Geophysicists, Jan. 2010, pp. 2773–2778.

[153] L. T. Rekanos, S. M. Panas, and T. D. Tsiboukis. "Microwave imaging using the finite-element method and a sensitivity analysis approach". In: *IEEE Trans. Med. Imaging* 18.11 (1999), pp. 1108–1114.

[154] R. J. Renka. "Nonlinear least squares and Sobolev gradients". In: *Appl. Numer. Math.* 65 (2013), pp. 91–104.

[155] W. B. Richardson. "High-order Sobolev preconditioning". In: *Nonlinear Anal. Theory, Methods Appl.* 63.5-7 (2005), pp. 1779–1787.

[156] E. Rohan and V. Lukeš. "Homogenization of the acoustic transmission through a perforated layer". In: *J. Comput. Appl. Math.* 234.6 (2010), pp. 1876–1885.

[157] P. Routh, R. Neelamani, R. Lu, et al. "Impact of high-resolution FWI in the Western Black Sea: Revealing overburden and reservoir complexity". In: *Lead. Edge* 36.1 (Jan. 2017), pp. 60–66.

[158] A. Royer, E. Béchet, and C. Geuzaine. "GMSH-FEM : AN EFFICIENT FINITE ELEMENT LIBRARY BASED ON GMSH". In: *14th World Congr. Comput. Mech. (WCCM), ECCOMAS Congr. 2020*. 2020.

[159] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Phys. D Nonlinear Phenom.* 60.1-4 (Nov. 1992), pp. 259–268.

[160] M. Sambridge and G. Drijkoningen. "Genetic algorithms in seismic waveform inversion". In: *Geophys. J. Int.* 109.2 (1992), pp. 323–342.

[161] G. Y. Sandhu, C. Li, O. Roy, S. Schmidt, and N. Duric. "Frequency domain ultrasound waveform tomography: breast imaging using a ring transducer". In: *Phys. Med. Biol.* 60.14 (July 2015), pp. 5381–5398.

[162] O. Scherzer and J. Weickert. "Relations between regularization and diffusion filtering". In: *J. Math. Imaging Vis.* 12.1 (2000), pp. 43–63.

[163] S. H. Schot. "Eighty years of Sommerfeld's radiation condition". In: *Hist. Math.* 19.4 (1992), pp. 385–401.

[164] M. K. Sen and P. L. Stoffa. "Nonlinear one-dimensional seismic waveform inversion using simulated annealing". In: *GEOPHYSICS* 56.10 (Oct. 1991), pp. 1624–1638.

[165] C. Shin and Y. H. Cha. "Waveform inversion in the Laplace domain". In: *Geophys. J. Int.* 173.3 (June 2008), pp. 922–931.

[166] C. Shin and W. Ha. "A comparison between the behavior of objective functions for waveform inversion in the frequency and Laplace domains". In: 73.5 (2008).

[167] C. Shin, N. H. Koo, Y. H. Cha, and K. P. Park. "Sequentially ordered single-frequency 2-D acoustic waveform inversion in the Laplace-Fourier domain". In: *Geophys. J. Int.* 181.2 (2010), pp. 935–950.

[168] C. Shin and D.-J. Min. "Waveform inversion using a logarithmic wavefield". In: *GEOPHYSICS* 71.3 (May 2006), R31–R42.

[169] C. Sideris, E. Garza, and O. P. Bruno. "Ultrafast Simulation and Optimization of Nanophotonic Devices with Integral Equation Methods". In: *ACS Photonics* 6.12 (Dec. 2019), pp. 3233–3240.

[170] S. Silver. *Microwave antenna theory and design*. McGraw-Hill, 1949.

[171] L. Sirgue and R. G. Pratt. "Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies". In: *GEOPHYSICS* 69.1 (Jan. 2004), pp. 231–248.

[172] F. Sourbier, A. Haidar, L. Giraud, et al. "Three-dimensional parallel frequency-domain visco-acoustic wave modelling based on a hybrid direct/iterative solver". In: *Geophys. Prospect.* 59.5 (Sept. 2011), pp. 834–856.

[173] T. Steihaug. "The Conjugate Gradient Method and Trust Regions in Large Scale Optimization". In: *SIAM J. Numer. Anal.* 20.3 (June 1983), pp. 626–637.

[174] D. Sun, K. Jiao, X. Cheng, and D. Vigh. "Compensating for source and receiver ghost effects in full waveform inversion and reverse time migration for marine streamer data". In: *Geophys. J. Int.* 201.3 (2015), pp. 1507–1521.

[175] H. Sun and L. Demanet. "Extrapolated full-waveform inversion with deep learning". In: *GEOPHYSICS* 85.3 (May 2020), R275–R288.

[176] J. H. Tang, R. Brossier, and L. Métivier. "Fully scalable solver for frequency-domain visco-elastic wave equations in 3D heterogeneous media: A controllability approach". In: *J. Comput. Phys.* 468 (2022), p. 111514.

[177] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Jan. 2005, pp. 1–358.

[178] A. Tarantola. "Inversion of seismic reflection data in the acoustic approximation". In: *GEOPHYSICS* 49.8 (Aug. 1984), pp. 1259–1266.

[179] J. Thurin, R. Brossier, and L. Métivier. "Ensemble-based uncertainty estimation in full waveform inversion". In: *Geophys. J. Int.* 219.3 (2019), pp. 1613–1635.

[180] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Dordrecht: Springer Netherlands, 1995.

[181] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. New York: Wiley, 1977.

[182] A.-M. Toader and C. Barbarosie. "6. Optimization of eigenvalues and eigenmodes by using the adjoint method". In: *Topol. Optim. Optim. Transp.* (2017), pp. 1–18.

[183] K. Torres and M. Sacchi. "Least-squares reverse time migration via deep learning-based updating operators". In: *GEOPHYSICS* 87.6 (Nov. 2022), S315–S333.

[184] D. A. Tortorelli and P. Michaleris. "Design sensitivity analysis: Overview and review". In: *Inverse Probl. Eng.* 1.1 (Oct. 1994), pp. 71–105.

[185] P. H. Tournier, I. Aliferis, M. Bonazzoli, et al. "Microwave tomographic imaging of cerebrovascular accidents by using high-performance computing". In: *Parallel Comput.* 85 (2019), pp. 88–97.

[186] P.-T. Trinh, R. Brossier, L. Métivier, L. Tavard, and J. Virieux. "Efficient time-domain 3D elastic and viscoelastic full-waveform inversion using a spectral-element method on flexible Cartesian-based mesh". In: *GEOPHYSICS* 84.1 (Jan. 2019), R61–R83.

[187] R. Versteeg. "The Marmousi experience: Velocity model determination on a synthetic complex data set". In: *Lead. Edge* 13.9 (Sept. 1994), pp. 927–936.

[188] J. Virieux and S. Operto. "An overview of full-waveform inversion in exploration geophysics". In: *GEOPHYSICS* 74.6 (2009).

[189] J. Virieux and S. Operto. "An overview of full-waveform inversion in exploration geophysics". In: *GEOPHYSICS* 74.6 (Nov. 2009), WCC1–WCC26.

[190] D. Voytan, A. Gomes, D. Datta, et al. "Mitigating elastic effects of acoustic full-waveform inversion with deep learning and application to field data". In: *Second Int. Meet. Appl. Geosci. Energy*. Society of Exploration Geophysicists and American Association of Petroleum Geologists, Aug. 2022, pp. 857–861.

[191] M. Warner and L. Guasch. "Adaptive waveform inversion: Theory". In: *GEOPHYSICS* 81.6 (Nov. 2016), R429–R445.

[192] M. Warner, A. Ratcliffe, T. Nangoo, et al. "Anisotropic 3D full-waveform inversion". In: *GEOPHYSICS* 78.2 (2013).

[193] J. Weickert. "Theoretical Foundations of Anisotropic Diffusion in Image Processing". In: (1996), pp. 221–236.

[194] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner Stuttgart, 1998.

[195] J. Weickert. "Coherence-enhancing diffusion of colour images". In: *Image Vis. Comput.* 17.3-4 (1999), pp. 201–212.

[196] P. Wellington, R. Brossier, S. Garambois, et al. "Laplacian based Preconditioning of FWI: Using prior information from seismic reflection data." In: *SEG Tech. Progr. Expand. Abstr. 2015*. Vol. 34. Society of Exploration Geophysicists, Aug. 2015, pp. 1436–1440.

[197] J. Wiskin, B. Malik, D. Borup, N. Pirshafiey, and J. Klock. "Full wave 3D inverse scattering transmission ultrasound tomography in the presence of high contrast". In: *Sci. Rep.* 10.1 (Dec. 2020), p. 20166.

[198] R. Wu and M. N. Toksöz. "Diffraction tomography and multisource holography applied to seismic imaging". In: *GEOPHYSICS* 52.1 (Jan. 1987), pp. 11–25.

[199] Z. Xue, H. Zhu, and S. Fomel. "Full-waveform inversion using seislet regularization". In: *Geophysics* 82.5 (2017), A43–A49.

[200] X. Yan, Q. He, and Y. Wang. "Truncated trust region method for nonlinear inverse problems and application in full-waveform inversion". In: *J. Comput. Appl. Math.* 404 (Apr. 2022), p. 113896.

[201] H. Yang, J. Jia, B. Wu, and J. Gao. "Mini-batch optimized full waveform inversion with geological constrained gradient filtering". In: *J. Appl. Geophys.* 152 (May 2018), pp. 9–16.

[202] P. Yang, J. Gao, and B. Wang. "A graphics processing unit implementation of time-domain full-waveform inversion". In: *GEOPHYSICS* 80.3 (2015), F31–F39.

[203] Y. Yang and B. Engquist. "Analysis of optimal transport and related misfit functions in full-waveform inversion". In: *GEOPHYSICS* 83.1 (Jan. 2018), A7–A12.

[204] J. Yao, L. Guasch, M. Warner, D. Davies, and A. Wild. "Removing Elastic Effects in FWI Using Supervised Cycled Generative Adversarial Networks". In: *81st EAGE Conf. Exhib. 2019 Work. Program.* European Association of Geoscientists & Engineers, 2019, pp. 1–5.

[205] H. Zhang, X. Li, H. Song, and S. Liu. "An adaptive subspace trust-region method for frequency-domain seismic full waveform inversion". In: *Comput. Geosci.* 78.February (May 2015), pp. 1–14.

[206] J. Zhang and M. N. Toksöz. "Nonlinear refraction traveltime tomography". In: *SEG Tech. Progr. Expand. Abstr. 1996*. Vol. 63. 5. Society of Exploration Geophysicists, Jan. 1996, pp. 2011–2014.

[207] W. Zhang and Y. Li. "Elastic wave full-waveform inversion in the time domain by the trust region method". In: *J. Appl. Geophys.* 197.May 2021 (Feb. 2022), p. 104540.

[208] H. Zhu, S. Li, S. Fomel, G. Stadler, and O. Ghattas. "A Bayesian approach to estimate uncertainty for full-waveform inversion using a priori information from depth migration". In: *GEOPHYSICS* 81.5 (Sept. 2016), R307–R323.

[209] L. Zhu, E. Liu, and J. H. McClellan. "Sparse-promoting full-waveform inversion based on online orthonormal dictionary learning". In: *GEOPHYSICS* 82.2 (2017), R87–R107.

[210] M. A. Zuberi and R. G. Pratt. "Mitigating nonlinearity in full waveform inversion using scaled-Sobolev pre-conditioning". In: *Geophys. J. Int.* 213.1 (Apr. 2017), pp. 706–725.