A penalty-free approach to PDE constrained optimization: Application to an inverse wave problem

Alexandre Hoffmann, Vadim Monteiller, Cédric Bellis

A penalty-free approach to PDE constrained optimization: Application to an inverse wave problem

Alexandre Hoffmann[1,2], Vadim Monteiller[1] and Cédric Bellis[1]

[1] Aix Marseille Univ, CNRS, Centrale Marseille, LMA UMR 7031, Marseille, France

[2] Univ Grenoble Alpes, ISTerre, Grenoble, France

E-mail: alexandre.hoffmann@univ-grenoble-alpes.fr

Contents

Abstract.

Inverse Wave Problems (IWPs) amount in non-linear optimization problems where a certain distance between a *state variable* and some observations of a wavefield is to be minimized. Additionally, we require the state variable to be the solution of a *model* equation that involves a set of *parameters* to be optimized. Typical approaches to solve IWPs includes the *adjoint method*, which generates a sequence of parameters and strictly enforces the model equation at each iteration, and, the *Wavefield Reconstruction Inversion (WRI)* method, which jointly generates a sequence of parameters and state variable but does not strictly enforce the model. WRI is considered to be an interesting approach because, by virtue of not enforcing the model at each iteration, it expands the search space, and can thus find solutions that may not be found by a typical adjoint method. However, WRI techniques generally requires the tuning of a penalty parameter until the model equation is considered satisfied. Alternatively, a fixed penalty parameter can be chosen but, in such case, it is impossible for the algorithm to find a solution that satisfies the model equation exactly.

In the present work, we present a, to our knowledge, novel technique of WRI type which jointly generates a sequence of parameters and state variable, and which *loosely* enforces the model. The method is based on a Trust Region-Sequential Quadratic Programming (TR-SQP) method which aims at minimizing, at each iteration, both the residual relative to the *linearized* model and a quadratic approximation of the cost functional. Our method approximately solves a sequence of quadratic subproblems by using a Krylov method. The Hessian-vector product is computed using the *second-order adjoint method*. The method is demonstrated on a synthetic case, with a configuration relevant to medical imaging.

1. Introduction

Inverse Wave Problems (IWPs) appear in applications that include seismic imaging, where it is commonly coined as Full Waveform Inversion (FWI) and aims at reconstructing the distribution of elasticity parameters within a certain domain in order to identify Earth's composition, and medical imaging, where one tries to image living tissues by characterizing, e.g., the bulk modulus and the mass density of a body. IWPs typically involve solving a highly non-linear and ill-posed optimization problem where one tries to minimize a certain *distance* d between *observed* wavefields u_{obs} in a subdomain of a domain Ω, and *simulated* wavefields $u_{\text{sim}}(p)$ that satisfy a *model equation* $\mathcal{M}(u_{\text{sim}}(p), p) = f$ in Ω, which involves a set of parameters p and an external source term f. With these notations, the typical IWP can be formulated as such:

$$\begin{aligned} &\underset{p \in \mathbb{P}}{\text{minimize}} && \hat{\mathcal{J}}(p) := d(u_{\text{sim}}(p), u_{\text{obs}}) \\ &\text{where} && \mathcal{M}(u_{\text{sim}}(p), p) = f, \end{aligned} \tag{1}$$

where the admissibility space $\mathbb{P}$ of parameters is to be specified. The functional $\hat{\mathcal{J}}$ will be designated as a *reduced cost functional* in the sequel. In the context of first-order gradient-based minimization approaches, differentiating u_{sim} with respect to p can be proven to be a convoluted task. However, it is possible to compute the gradient of

$\hat{\mathcal{J}}$ by using the so-called *adjoint method*, which was first introduced in the context of optimal control theory [16], then applied to parameter identification [17] and later to weather forecasting [18]. The adjoint method is also widely used in seismic imaging, see Plessix [19] for an overview. Moreover, it is also possible to compute the product between a direction $\tilde{p}$ and the Hessian of $\hat{\mathcal{J}}$ by using the second-order adjoint method [20, 22, 21]. It is thus possible to solve (1) with standard optimization methods, see [7, Chapters 6, 8 and 9] for more details. By following the adjoint method, it is possible to compute the gradient of $\hat{\mathcal{J}}$ by solving two Partial Differential Equations (PDEs) and to compute the product between a direction $\tilde{d}$ and the Hessian operator by solving two additional PDEs. Hence some prefer to use quasi-Newton methods, which rely on inexpensive approximations of the Hessian, instead. Alternatively, Van Leeuwen and Herrmann proposed to relax the model equation [23]. More precisely, u_{sim} is not chosen as the solution of the model equation but is rather free. Thus, this approach considers both the wavefield u and the parameter p as degrees of freedom. In the remainder of this article, the search space of such method is denoted as $\mathbb{X} := \mathbb{U} \times \mathbb{P}$, which is commonly referred to as an "extended domain". More precisely, Van Leeuwen and Herrmann proposed to *loosely* enforce the model equation by adding a *penalty function* [23]. Such method can be referred to as the Wavefield Reconstruction Inversion (WRI) method and it can be written as the following optimization problem:

$$\underset{(u,p)\,\in\,\mathbb{U}\times\mathbb{P}}{\text{minimize}} \quad \Phi(u,p) := d(u,u_{\text{obs}}) + \sigma\; g\left(\mathcal{M}(u,p) - f\right), \tag{2}$$

where, in addition to $\mathbb{P}$, the state space $\mathbb{U}$ is also to be specified. In (2), the term g is often referred to as the *exterior penalty function* and σ is referred to as the *penalty parameter*. Van Leeuwen and Herrmann then extended their method to general PDE constrained optimization problem [24], around the same time, a "discretize then optimize" penalty-based method for general PDE constrained optimization problems was proposed by Kaltenbacher [32], and, more recently, Aghamiry *et al* proposed a similar method [26] specifically for the FWI problem. Van Leeuwen and Herrmann's method use Newton's method and a linesearch to update the parameter p. It proceeds to update the wavefield u by solving an optimization problem. The stopping criterion of Van Leeuwen and Herrmann's method relies on the norm of the Lagrange function associated with (1). Thus, evaluating the stopping criterion requires to compute a Lagrange multiplier λ, which is done by computing the residual of the model PDE. A new descent direction along p is then computed from both u and λ. Whereas Aghamiry *et al* 's method updates both the wavefield u and by solving two optimization problems and then the parameter p by solving another optimization problem. Penalty-based methods typically require to start with a relatively low value σ and to increase it after each minimization. However, Aghamiry *et al* choose a certain σ and do not update it. A compromise is thus made between minimizing the misfit between the observed and the simulated data and between satisfying the model equation. On the other hand, Van Leeuwen and Herrmann proposes to increase the penalty parameter when the model residual becomes too large. Hence the model equation is always solved upon convergence.

In the present work, we choose a different road. We first reformulate IWPs as a constrained optimization problem. We solve this optimization problem using the Trust Region-Sequential Quadratic Programming (TR-SQP) method. TR-SQPs methods have been applied to PDE constrained problems that arise in various fields of physics. Most of them with a "discretize then optimize" approach, see [33, 36, 10, 35, 34], in which the PDE constrained problem is first turned into a finite dimensional constrained optimization problem and then solved. Alternatively, Ziems and Ulbrich proposed a TR-SQP method that discretizes the problem first but solves it with techniques that do not require any prior discretization [11]. However, such methods have never been applied to FWI problems. Moreover, the method we propose is a fully "optimize then discretize" method, which means that each step is derived as a function and then sampled on a grid. Our method generates a joint sequence of parameters p, of simulated data u_{sim} and of adjoint wavefields that, at convergence, satisfy the Karush-Kuhn-Tucker (KKT) conditions of the constrained optimization problem. However, our method does not strongly enforce the model equation at each iteration. Instead, it *loosely enforces* a linearized model equation. Our method requires to solve a sequence of optimization problem with a quadratic cost functional, a linearized model constraint and a Trust Region (TR) constraint. This problem is solved by using the second-order adjoint method. Our method presents the advantage of working in an extended search space, while being *globally convergent* thanks to the TR constraint and to converge towards *a* solution of the KKT system without the need for the adjustment of a penalty parameter. We should underline that "globally convergent" is understood in the sense that the method will reach a local minimum regardless of the starting condition. It does not ensure that the method will reach the so-called *global minimum* nor the *true* parameter.

This paper is organized as follows. First, we introduce the PDE constrained optimization problem considered for a generic model equation in Section 2 and describe the TR-SQP method. The proposed approach is described in Section 3 and we then proceed in Section 4 to give an application to an IWP, for which the model equation is the scalar Helmholtz equation. Numerical results are finally presented and discussed in Section 4.3.

2. Preliminaries

2.1. PDE constrained optimization problem

We start by rewriting the IWP as a constrained optimization problem where, for the sake of notation, the state variable $u \in \mathbb{U}$ and the parameters $p \in \mathbb{P}$ have been gathered into a single variable $x := (u, p) \in \mathbb{X} := \mathbb{U} \times \mathbb{P}$, with $\mathbb{U}$ and $\mathbb{P}$ being some Hilbert spaces. Let $\mathcal{J} : \mathbb{U} \to \mathbb{R}$ be our cost functional, $\mathcal{M} : \mathbb{X} \to \mathbb{V}^*$ be our model equation, considering a dual space $\mathbb{V}^*$ (in most applications $\mathbb{V} = \mathbb{U}$), and $f \in \mathbb{V}^*$ be our source term. Moreover, the model equation is typically given in a weak form as:

$$\langle \mathcal{M}(x) - f, v \rangle_{\mathbb{V}^*,\mathbb{V}} = a(u, v; h(p)) - \ell(v; f) = 0,$$

where the duality product $\langle \cdot, \cdot \rangle_{\mathbb{V}^*,\mathbb{V}}$ writes in terms of a bilinear form a and a linear form ℓ, which depend on the parameters p and source f, respectively. Moreover, h is a function that ensures the positivity of the reconstructed parameters ‡. With this formalism, the IWP can be written as the following constrained optimization problem:

$$\underset{x \in \mathbb{X}}{\text{minimize}} \quad \mathcal{J}(x) \tag{3a}$$

$$\text{subject to} \quad \mathcal{M}(x) = f \ . \tag{3b}$$

The Lagrangian function $\mathcal{L} : \mathbb{X} \times \mathbb{V}^{**} \to \mathbb{R}$ corresponding to (3) can be written as such:

$$\mathcal{L}(x, \lambda) = \mathcal{J}(x) + \langle \mathcal{M}(x) - f, \lambda \rangle_{\mathbb{V}^*,\mathbb{V}}, \tag{4}$$

with $\mathbb{V}^{**} = \mathbb{V}$ so that $\lambda \in \mathbb{V}$. Note that, until now we only considered real-valued state variable u. In some cases, the governing PDE for our wavefiled is complex-valued. In such case, we choose to consider the real and imaginary part of our wavefiled as two separated state variables. Thus, our model equation, $\mathcal{M}$, may involve a coupling between the part of u that represents the real part of our wavefiled and between the part of u that represents the imaginary part of our wavefiled. In such case, if we need to solve the model equation, we can either solve a coupled PDE or solve a complex-valued PDE with a complex-valued solution $\hat{u}$. In practice, for the sake of performance, we chose the latter option. The state variable u will then be retrieved from the real and imaginary part of $\hat{u}$.

2.2. TR-SQP approach

TR-SQP methods typically attempt to solve a non-linear optimization problem by solving a sequence of Quadratic Programs (QPs) with both linear constraints and a TR constraint. Each step is computed by solving a QP, referred to as subproblems, and then evaluated by means of a *merit function*. If the step is good enough it is accepted, while if it is not, then it is rejected and the TR radius is reduced. This section first describes how to derive the TR-SQP subproblems. We then present a method that allows us to solve each subproblem by solving a series of linear PDEs. Finally we present the merit function we use in this work.

2.2.1. TR-SQP subproblems. The Sequential Quadratic Programming (SQP) method consists in applying Newton's method to the KKT optimality conditions of (3), with steps $dx = (x_{k+1} - x_k)$ and $d\lambda = (\lambda_{k+1} - \lambda_k)$, which, after gathering some terms, leads to the following system of equations:

$$D\mathcal{J}(x_k) + D\mathcal{M}^*(x_k)\lambda_{k+1} + D^2_{xx}\mathcal{L}(x_k, \lambda_k)dx = 0, \tag{5a}$$

$$\mathcal{M}(x_k) - f + D\mathcal{M}(x_k)dx = 0. \tag{5b}$$

‡ Numerically, we do not need to ensure that $h(p)$ is bounded from above. Hereafter, we thus choose $h(p) = \exp(p)$. However, one can easily bound p both from above and from below by using, e.g., an arc-tangent function.

Note that in (5), one has $D\mathcal{J}(x) \in \mathbb{X}^*$ and $D\mathcal{M}(x) : \mathbb{X} \to \mathbb{V}^*$ with adjoint $D\mathcal{M}^*(x) : \mathbb{V} \to \mathbb{X}^*$. Moreover, the second-order derivative satisfies $D^2_{xx}\mathcal{L} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$, so that $D^2_{xx}\mathcal{L}(x_k, \lambda_k)dx \in \mathbb{X}^*$. The key point of the SQP method is to notice that, at a given iterate k, equations (5) correspond to the optimality conditions of an optimization problem with a quadratic cost functional and a linear constraint. TR-SQP methods add an additional bound constraint on the variable dx to ensure that it is always possible (if the model constraint and the TR constraint are compatible, i.e. the feasible set of (5) is not empty) to find a bounded solution to the subproblem at each iterate (x_k, λ_k). If both constraints are not compatible, then Nocedal and Wright [7, Chapter 18.8], among others, propose to relax the model equation. Each subproblem can thus be written as such:

$$\begin{aligned} \underset{dx \in \mathbb{X}}{\text{minimize}} \quad & \frac{1}{2}\big\langle D^2_{xx}\mathcal{L}(x_k, \lambda_k)dx, dx\big\rangle_{\mathbb{X}^*,\mathbb{X}} + \langle D\mathcal{J}(x_k), dx\rangle_{\mathbb{X}^*,\mathbb{X}} && \text{(6a)} \\ \text{subject to} \quad & \mathcal{M}(x_k) + D\mathcal{M}(x_k)dx = f, && \text{(6b)} \\ & \mathcal{N}(dx) \leq \Delta_k, && \text{(6c)} \end{aligned}$$

with λ_{k+1} in (5) corresponding to the Lagrange multiplier of (6), $\mathcal{N}$ being a norm on $\mathbb{X}$ and $\Delta_k \in \mathbb{R}$ being the TR radius that will be updated depending on *how good of a step* we find by solving (6). The process through which we determine if a step is *good* or not will be described later on. Note finally that (6) may have an empty feasible set. This problem can be overcome with an array of techniques (for an overview, see [7, Chapter 18]). In the present work, we choose the Byrd-Omojokun approach [8], which has proven useful in the context of PDE-constrained optimization [33, 36, 10, 35, 34, 11].

2.2.2. Computing a TR-SQP step. The Byrd-Omojokun approach consists in splitting the step dx into two steps, namely the quasi-normal step dx_n and the tangential step dx_t, as $dx = dx_n + dx_t$. Each of these steps are computed by solving a well posed QP. We first compute dx_n, a step that improves the feasibility of our solution. Such a step is referred to as the *quasi-normal* step towards feasibility and is computed as the solution of the following subproblem:

$$\begin{aligned} \underset{dx_n \in \mathbb{X}}{\text{minimize}} \quad & \|\mathcal{M}(x_k) + D\mathcal{M}(x_k)dx_n - f\|_{\mathbb{V}^*} && \text{(7a)} \\ \text{subject to} \quad & \mathcal{N}(dx_n) \leq \zeta\Delta_k. && \text{(7b)} \end{aligned}$$

where $\zeta \in (0, 1)$ ensures that dx_n is neither too small or too large relatively to dx_t [7, Chapter 18.8]. We then compute dx_t, a step that improves the optimality of our solution. This step is typically called the *tangential* step towards optimality and, assuming that the solution to (7) has been found, then dx_t is computed as the solution of the following

subproblem:

$$\underset{dx_t \in \mathbb{X}}{\text{minimize}} \quad Q_k(dx_t) := \frac{1}{2}\left\langle D^2_{xx}\mathcal{L}(x_k,\lambda_k)dx_t, dx_t\right\rangle_{\mathbb{X}^*,\mathbb{X}} + \left\langle D\mathcal{J}(x_k) + D^2_{xx}\mathcal{L}(x_k,\lambda_k)dx_n, dx_t\right\rangle_{\mathbb{X}^*,\mathbb{X}} \tag{8a}$$

$$\text{subject to} \quad D\mathcal{M}(x_k)dx_t = 0, \tag{8b}$$

$$\mathcal{N}(dx_t)^2 \leq \Delta_k^2 - \mathcal{N}(dx_n)^2. \tag{8c}$$

Note that, dx_n is called the quasi-normal step because, if both the TR constraint (6c) and the linearized constraint (6b) are compatible, then dx_n is the orthogonal projection of the center of the TR onto the range of the linearized model equation. While the tangential step dx_t lies within the null-space of the linearized model equation. Thus, if computed properly, dx_n and dx_t are orthogonal in the sense of the above. This is why, the TR constraint (8c) is written as a sum between the norm of dx_n and the norm of dx_t. Owing to the decomposition of x into u, the state variable, and p a set of model parameters, and following [33, 36, 10, 35, 34, 11], we choose the following splitting:

$$dx_n = (du_n, 0) \qquad \text{and} \qquad dx_t = (du_t, dp). \tag{9}$$

The advantage of such a splitting is that we usually have a solver for the linear model $D_u\mathcal{M}(u,p)$ to be used in (7). We can indeed compute a sufficiently good quasi-normal step (i.e. a quasi-normal step that guarantees basic global convergence of the method) by solving the following PDE:

$$D_u\mathcal{M}(u_k,p_k)du_n = -(\mathcal{M}(u_k,p_k) - f) \tag{10}$$

and then scale the solution such that it satisfies (7b), see e.g. [9, 10, 11]. This allows us to compute a satisfying enough approximate solution to (7) by simply solving a PDE. Moreover, with (9), the tangential subproblem (8) can be written as a PDE constrained optimization problem with the additional TR constraint (8c).

2.3. Evaluating a TR-SQP step

In the previous section, we have presented a way to solve the Trust Region Subproblems (TRSs) for a given TR radius Δ_k. However, Trust Region Methods typically requires us to evaluate whether a step dx is acceptable or not. For unconstrained problem, the goodness of a step k is evaluated by comparing the decrease of the quadratic approximation of the cost functional, i.e. the *predicted reduction* $\text{pred}_k(dx)$, to the decrease of the cost functional, i.e. the *actual reduction* $\text{ared}_k(dx)$. If the ratio $\rho_k := \frac{\text{ared}_k(dx)}{\text{pred}_k(dx)}$ is sufficiently large, the step is accepted (i.e. $x_{k+1} = x_k + dx$) and Δ_k may be increased. However, if ρ_k is to small, the step is rejected and Δ_k is decreased. The computation of $\text{ared}_k(dx)$ and of $\text{pred}_k(dx)$ are detailed in Appendix B.

Algorithm 1: TR-SQP

– *Initialization:*

Define first guess $x_0 = (u_0, p_0)$ and residual error ϵ
Estimate λ_0
Set Δ_0
Set $\rho_0 = 0$, $\mu_0 = 1$ and choose thresholds $\rho_{\min}$ and $\rho_{\max}$
Choose two constants $\gamma_1 < 1 < \gamma_2$

– *Then:* iterate until $\|\nabla\mathcal{L}(u_k, p_k, \lambda_k)\|_{\mathbb{X}\times\mathbb{V}} \leq \epsilon$

I. Repeat
  1. Compute a quasi-normal step $dx_n = (du_n, 0)$ by approximately solving (10)
  2. Compute a tangential step $dx_t = (du_t, dp)$, see Algorithm 2
  3. Evaluate $dx = dx_n + dx_t$
  4. Evaluate the ratio $\rho_k = \text{ared}_k(dx)/\text{pred}_k(dx)$ from (B.2) and (B.3-B.4)
  5. If $\rho < \rho_{min}$ then $\Delta_k \leftarrow \gamma_1\Delta_k$
  6. Else
    a. If $\rho \in [\rho_{\min}, \rho_{\max}]$ then $\Delta_{k+1} = \Delta_k$
      Else if $\rho > \rho_{\max}$ then $\Delta_{k+1} = \gamma_2\Delta_k$
    b. Break

II. Set $x_{k+1} = x_k + dx$

III. Update Lagrange multiplier λ_{k+1}, see Section 3.3.2

2.4. Overview of TR-SQP and objectives

Let us now give, in Algorithm 1, a summary of the TR-SQP worklfow.

Remark 1 *Note that ρ_{min} and ρ_{max} are some user-chosen thresholds that allow to evaluate how good is a step. Typically one sets $\rho_{min} = 0.25$ and $\rho_{max} = 0.75$. Moreover, γ_1 and γ_2 are also some user-chosen constants that dictate how to shrink and expand the TR radius Δ. Typical values for these two constants are $\gamma_1 = 0.25$ and $\gamma_2 = 2$. In most application, $\Delta_0 = 1$. However, if for some reasons, the norm of the solution x^* to* (3) *is expected to be very large, because of a very large source term or a very large computational domain for example, then Δ_0 should be scaled appropriately for a faster convergence, for example by setting $\Delta_0 = \|x_0\|_{\mathbb{X}}$.*

Remark 2 *In general, the choice of x_0 is not crucial to the algorithm performances. However, in the case of PDE constrained optimization, a natural choice for u_0 is the solution of the model equation* (3b) *given p_0. The choice of p_0 will highly depend on the type of applications. As an example, in medical imaging p_0 likely is chosen to be homogeneous. However, in seismic tomography a better first guess is often required. A natural choice for λ_0 is the solution of the adjoint equation:*

$$D_u\mathcal{M}^*(u_0, p_0)\lambda_0 = -D_u\mathcal{J}(u_0, p_0).$$

Note that with such an initialization, the derivative in p of the Lagrange function (4) *is equal to the derivative of the reduced cost functional* (1)*. However, after one iteration, there is no reason to expect that both derivatives would be equal.*

At step I.2 of Algorithm 1, we propose in the present article a, to our knowledge, new way to compute the tangential step dx_t that (i) does not require to approximate the second order derivatives of Q_k , (ii) nor rely on any prior discretization of the problem considered. In this framework, our primary objective is to apply, as far as we know, for the first time the TR-SQP method to the IWP with wavefield reconstruction. Our secondary objective is to demonstrate that, by constraining the L^2 norm of the gradient of dp, we can achieve satisfactory results using the proposed approach even if we do not have access to low frequency data.

3. Proposed method

3.1. Functional setting

In the present work, the norm on $\mathbb{X} = \mathbb{U} \times \mathbb{P}$ to be used in (6) to define the TR radius is defined as

$$\mathcal{N} : x = (u, p) \mapsto \mathcal{N}(x) = \left(\|u\|^2_{L^2(\Omega)} + \|\nabla p\|^2_{L^2(\Omega)}\right)^{1/2}. \tag{11}$$

Accordingly, a suited functional framework *with WRI* which corresponds to the choice of the Hilbert spaces $\mathbb{U} = \mathbb{V} = H^1_0(\Omega)^n$ and $\mathbb{P} = \left(H^1_0(\Omega) \cap L^\infty(\Omega)\right)^m$, in dimensions $n, m \in \mathbb{N}$ to be specified. Note that the choice for $\mathbb{P}$ arises from the chosen norm used for the TR constraint. This choice is rather pragmatic and motivated by the satisfying reconstruction results it yields numerically, an issue that will be discussed in Section 4.3.

3.2. Computing a TR-SQP step

Let us first recall that the tangential subproblem (8) can be written as a PDE constrained optimization problem with two variables, du_t a wavefield and dp a set of parameters. With such notations, (8) can be written as such:

$$\begin{aligned}
\underset{du_t \in \mathbb{U},\, dp \in \mathbb{P}}{\text{minimize}} \quad & Q_k(du_t, dp) && \text{(12a)}\\
\text{subject to} \quad & D_u\mathcal{M}(u_k, p_k)du_t = -D_p\mathcal{M}(u_k, p_k)dp, && \text{(12b)}\\
& \|du_t\|^2_{L^2(\Omega)} + \|\nabla dp\|^2_{L^2(\Omega)} \leq \Delta_k^2 - \|du_n\|^2_{L^2(\Omega)}, && \text{(12c)}
\end{aligned}$$

where Q_k is a quadratic functional that can be rewritten as such:

$$\begin{aligned}
Q_k(du_t, dp) = {} & \frac{1}{2}\left\langle D^2_{uu}\mathcal{L}(u_k, p_k, \lambda_k)du_t, du_t\right\rangle_{\mathbb{U}^*,\mathbb{U}} + \frac{1}{2}\left\langle D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k)dp, du_t\right\rangle_{\mathbb{U}^*,\mathbb{U}}\\
& + \frac{1}{2}\left\langle D^2_{pu}\mathcal{L}(u_k, p_k, \lambda_k)du_t, dp\right\rangle_{\mathbb{P}^*,\mathbb{P}} + \frac{1}{2}\left\langle D^2_{pp}\mathcal{L}(u_k, p_k, \lambda_k)dp, dp\right\rangle_{\mathbb{P}^*,\mathbb{P}}\\
& + \left\langle D_u\mathcal{J}(u_k, p_k) + D^2_{uu}\mathcal{L}(u_k, p_k, \lambda_k)du_n, du_t\right\rangle_{\mathbb{U}^*,\mathbb{U}}\\
& + \left\langle D_p\mathcal{J}(u_k, p_k) + D^2_{pu}\mathcal{L}(u_k, p_k, \lambda_k)du_n, dp\right\rangle_{\mathbb{P}^*,\mathbb{P}}.
\end{aligned}$$

Note that, unlike in the original problem (3), the model-based constraint (12b) is linear in (du_t, dp). More precisely, dp can be considered as the source term of a PDE system whose solution is du_t. A standard technique to solve this problem is to introduce a linear *solution operator* S such that $du_t = S(dp)$, i.e. $dx_t = (S(dp), dp)$ satisfies the model equation (12b). Note that, by introducing said solution operator, the TR constraint (12c) may be relaxed. Constraining the norm of $S(dp)$ can prove to be tricky, yet we can assume without loss of generality that S is a bounded operator, i.e., in accordance with (11), there exists a constant $\alpha > 0$ so that

$$\|S(dp)\|^2_{L^2(\Omega)} \leq \alpha \|\nabla dp\|^2_{L^2(\Omega)}.$$

More intuitively, the model equation (12b) ensures that if the norm of dp decreases, then the norm of $S(dp)$ decreases too. Note that α should be adjusted on the fly. Indeed, if left unchecked, the norm of du_t could be too large compared to du_n, which could be detrimental to the convergence of the method. For the above mentioned reasons, we rewrite our tangential subproblem using a reduced cost functional $\hat{Q}_k : \mathbb{P} \to \mathbb{R}$ as:

$$\underset{dp \in \mathbb{P}}{\text{minimize}} \quad \hat{Q}_k(dp) := Q_k(S(dp), dp) \tag{13a}$$

$$\text{subject to} \quad \|\nabla dp\|^2_{L^2(\Omega)} \leq \tilde{\Delta}_k^2 := \frac{\Delta_k^2 - \|du_n\|^2_{L^2(\Omega)}}{(1+\alpha)}. \tag{13b}$$

Let us first recall that, Q_k is a quadratic functional. Moreover, the solution operator S is linear. This means that $\hat{Q}_k$ can be written as follows:

$$\hat{Q}_k(dp) = \frac{1}{2}(dp, H_k dp)_{\mathbb{P}} + (g_k, dp)_{\mathbb{P}},$$

with $(\cdot,\cdot)_{\mathbb{P}}$ being an inner product on $\mathbb{P}$, in our case, the $L^2(\Omega)$ inner product, and where g_k is the gradient of the reduced functional $\hat{Q}_k$ evaluated at zero, while H_k is its Hessian, which is constant in dp. While the computation of the gradient and Hessian of Q_k is straightforward, this of g_k and, especially, H_k can be cumbersome because the solution operator S enters the definition of $\hat{Q}_k$. Thankfully, the gradient of the reduced functional can be computed by using the adjoint method, and the product between a direction dp and H_k can be computed with the so-called *second-order adjoint method* [22, 21]. In optimization, problems such as (13) are often labeled as TRSs. This sort of problems have been extensively studied (at least in finite dimensions) and various methods have been developed to solve them, see e.g. [7, 1]. Let us recall that, in our case, we do not want to compute explicitely the Hessian operator H_k. A method that only rely on computing the product between H_k and a direction $d \in \mathbb{P}$ is thus required to solve the reduced tangential subproblem (13). Such methods includes the Truncated Conjugate Gradient (TCG) method [4] and the Generalized Lanczos Trust Region (GLTR) method [2]. The TCG simply performs Conjugate Gradient (CG) iterations until a negative curvature direction is found or until the TR constraint is no longer satisfied. In such case, the TCG computes an approximate solution on the boundary of the TR by using

the current descent direction. The GLTR uses Lanczos iterations to project the TRS onto a finite dimensional Krylov subspace in which the Hessian is a small tri-diagonal matrix. In such a subspace, it is possible to use the Moré-Sorensen method [6] for a relatively low cost. Gould *et al* also note that the Lanczos iterates can be retrieved from CG iterations. It is thus possible to use the CG iterations, as long as they do not breakdown and then switch to the Moré-Sorensen method [2].

We should mention that, in order to ensure the global convergence of the TR-SQP, the tangential step must provide of fraction of the *Cauchy decrease condition* associated with the TRS (13), see [10, 11]:

$$\hat{Q}_k(0) - \hat{Q}_k(dp) \geq \kappa_1 \|g_k\|_{\mathbb{P}} \min\left\{\kappa_2 \|g_k\|_{\mathbb{P}}, \kappa_3 \tilde{\Delta}_k\right\} \tag{14}$$

where κ_1, κ_2 and κ_3 are positive constants independent of k. Using a reasoning similar to that of Heinkenschloss *et al* [10], we note that the GLTR method generates a sequence of steps $\{dp_k\}_{k=0\ldots k_{max}}$, where dp_k are spawned by a k-order Krylov subspace. Hence, by construction, dp_0 *is* the Cauchy point and thus satisfies the Cauchy decrease condition. Furthermore, the sequence $\{\hat{Q}_k(dp_k)\}_{k=0\ldots k_{max}}$ is nonincreasing, see [3, Theorem 4.6]. Hence, the dp computed with the GLTR satisfies (14) and is thus compatible with the convergence condition for the TR-SQP method.

By using the adjoint method (see Appendix A for more details), we find that the gradient of the reduced functional $\hat{Q}_k$ can be written as such:

$$D\hat{Q}_k(dp) = D_{dp}Q_k(S(dp), dp) + D_p\mathcal{M}^*(u_k, p_k)S_{\text{adj}}(dp), \tag{15}$$

where $S(dp)$ and $S_{\text{adj}}(dp)$ correspond respectively to the solutions of the following PDEs:

$$D_u\mathcal{M}(u_k, p_k)S(dp) + D_p\mathcal{M}(u_k, p_k)dp = 0, \tag{16a}$$

$$D_u\mathcal{M}^*(u_k, p_k)S_{\text{adj}}(dp) + D_{S(dp)}Q_k(S(dp), dp) = 0 \tag{16b}$$

and where $\mathcal{M}^*$, the adjoint model equation, and $D_p\mathcal{M}(u_k, p_k)$ the derivative of the model equation with respect to p, are defined below. Note that, when using the TCG and the GLTR methods, we need to compute the gradient of the reduced functional $\hat{Q}_k$ only at zero, which leads to some simplifications:

$$g_k := \nabla\hat{Q}_k(0) = \nabla_{dp}Q_k(0, 0) + \nabla_p\left[\mathcal{M}(u_k, p_k)S_{\text{adj}}(0)\right], \tag{17}$$

with the following abuse of notation:

$$\left(\nabla_p\left[\mathcal{M}(u_k, p_k)S_{\text{adj}}(0)\right], \tilde{p}\right)_{\mathbb{P}} = \langle D_p\mathcal{M}^*(u_k, p_k)S_{\text{adj}}(dp), \tilde{p}\rangle_{\mathbb{P}^*,\mathbb{P}} \qquad \forall\tilde{p} \in \mathbb{P}.$$

It is thus unnecessary to solve the state equation (16a). Now, by using the second-order adjoint method (see Appendix A for more details), we find that the product between a direction $d \in \mathbb{P}$ and the Hessian H_k of the reduced functional $\hat{Q}_k$ can be written as:

$$H_k d = \nabla_p\left[\mathcal{M}(u_k, p_k)\mu_{\text{adj}}\right] + \nabla^2_{dp\,du_t}Q_k(du_t, dp)\mu + \nabla^2_{dp\,dp}Q_k(du_t, dp)d, \tag{18}$$

where $\nabla^2_{dp\,du_t} Q_k(du_t, dp)$ is a linear application from $\mathbb{U}$ to $\mathbb{P}$ defined such that $\nabla^2_{dp\,du_t} Q_k(du_t, dp)\mu$ is the Riesz representant of $D^2_{dp\,du_t} Q_k(du_t, dp)\mu$ for all $\mu \in \mathbb{U}$, and likewise for $\nabla^2_{dp\,dp} Q_k(du_t, dp)$ and where μ and μ_{adj} are solution of the two following equations:

$$D_u \mathcal{M}(u_k, p_k)\mu = - D_p \mathcal{M}(u_k, p_k) d, \tag{19}$$

$$D_u \mathcal{M}^*(u_k, p_k)\mu_{\text{adj}} = - D^2_{du_t\,dp} Q_k(du_t, dp) d - D^2_{du_t\,du_t} Q_k(du_t, dp)\mu. \tag{20}$$

Note that, since Q_k is a quadratic functional, its second derivatives are constant with respect to du and dp. Hence, since the solution operator S is linear, the product between a direction d and the Hessian of the reduced cost functional $\hat{Q}_k$ is constant in dp.

3.3. Summary of the proposed method

3.3.1. Computation of the tangential step. The proposed method proceeds as follows:

Algorithm 2: Step I.2 of Algorithm 1
Computation of the tangential step $dx_t = (du_t, dp)$

I.2.a *Initialization of the GLTR method: compute the gradient of $\hat{Q}_k$ at zero.*
- (i) Compute $S_{\text{adj}}(0)$ by solving (16b)
- (ii) Compute g_k according to (17)

I.2.b *Each iteration of the GLTR method: Apply the Hessian of $\hat{Q}_k$ to $d \in \mathbb{P}$.*
- (i) Compute μ as the solution of (19)
- (ii) Compute μ_{adj} as the solution of (20)
- (iii) Compute $H_k d$ according to (18)

I.2.c *After convergence of the GLTR method: Let dp be an approximate solution to* (13)
- (i) Compute $S(dp)$ by solving (16a)
- (ii) Return $dx_t = (S(dp), dp)$

In Section 4, will see how it is implemented on an IWP.

3.3.2. Updating the Lagrange multiplier. At step III of Algorithm 1, the new Lagrange multiplier λ_{k+1} should ideally satisfy (5a). However, rewriting (5a) in (du, dp) makes it clear that it amounts in the following over-determined system:

$$D_u \mathcal{J}(u_k, p_k) + D_u \mathcal{M}^*(u_k, p_k)\lambda_{k+1} + D^2_{uu}\mathcal{L}(u_k, p_k, \lambda_k) du + D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k) dp = 0, \tag{21a}$$

$$D_p \mathcal{J}(u_k, p_k) + D_p \mathcal{M}^*(u_k, p_k)\lambda_{k+1} + D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k) du + D^2_{pp}\mathcal{L}(u_k, p_k, \lambda_k) dp = 0. \tag{21b}$$

For finite dimensional problems, λ_{k+1} is typically computed as the solution to (21). However, when dealing with infinite dimensional problems this requires to minimize the sum of squared norms of operators, which is not a trivial task. It is possible to

circumvent this by choosing λ_{k+1} as $S_{\text{adj}}(0)$, see [11]. We posit that $S_{\text{adj}}(dp)$, with dp being the solution to the tangential subproblem (13) is a better choice. Indeed, since $du = du_n + S(dp)$, we can rewrite (16b) as such:

$$D_u\mathcal{J}(u_k, p_k) + D_u\mathcal{M}^*(u_k, p_k)S_{\text{adj}}(dp) + D^2_{uu}\mathcal{L}(u_k, p_k, \lambda_k)du + D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k)dp = 0.$$

Similarly, (15) can, be written as such:

$$D_p\mathcal{J}(u_k, p_k)+D_p\mathcal{M}^*(u_k, p_k)S_{\text{adj}}(dp)+D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k)du+D^2_{pp}\mathcal{L}(u_k, p_k, \lambda_k)dp = D\hat{Q}_k(dp).$$

Thus, choosing λ_{k+1} as $S_{\text{adj}}(dp)$ ensures that (21a) is exactly solved and that the error we commit in (21b) is $\|D\hat{Q}_k(dp)\|_{\mathbb{P}^*}$ as opposed to $\|D^2_{uu}\mathcal{L}(u_k, p_k, \lambda_k)S(dp) + D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k)dp\|_{\mathbb{U}^*} + \|D^2_{up}\mathcal{L}(u_k, p_k, \lambda_k)S(dp) + D^2_{pp}\mathcal{L}(u_k, p_k, \lambda_k)dp - D\hat{Q}_k(dp)\|_{\mathbb{P}^*}$ if we had chosen λ_{k+1} as $S_{\text{adj}}(0)$. This is interesting because, since the norm of $D\hat{Q}_k(dp)$ is used as a convergence criterion when solving (13), it is expected to be small if we have successfully solved the tangential subproblem (13).

4. Application example to an IWP

4.1. Setting

In the present work, we apply our method to an IWP in acoustics and for a two-dimensional configuration that relates in particular to medical imaging. More precisely, we consider an infinite domain in which an object to be reconstructed is placed. The medium is characterized by its mass density ρ and bulk modulus κ. The said object is illuminated by a collection of external harmonic sources $\{\hat{f}^{\ell s}\}$, at a set of frequencies $\{\omega_\ell\}$ and locations $\{\boldsymbol{x}_s\}$, each giving rise to an observable acoustic pressure field $\hat{u}^{\ell s}_{\text{obs}}$ that is governed by the following PDE:

$$-\nabla\cdot\left(\frac{1}{\rho(\boldsymbol{x})}\nabla\hat{u}^{\ell s}_{\text{obs}}(\boldsymbol{x})\right) - \omega^2_\ell\frac{1}{\kappa(\boldsymbol{x})}\hat{u}^{\ell s}_{\text{obs}}(\boldsymbol{x}) = \hat{f}^{\ell s} \qquad \forall\boldsymbol{x}\in\mathbb{R}^2,$$

which is typically obtained by taking the Fourier transform of the time-domain wave equation, and satisfies the Sommerfeld radiation condition. Note that, in most applications, the source term can be written as follows:

$$\hat{f}^{\ell s} = a(\omega_\ell)\delta(\boldsymbol{x}-\boldsymbol{x}_s),$$

for point sources at $\{\boldsymbol{x}_s\}$ with amplitude $a(\omega_\ell)$. Moreover, in medical imaging, the sources can generally be placed around the Region of Interest (ROI), while in seismic imaging they are constrained to be located at the top of the domain. Also note that, instead of working with complex-valued pressure fields, we introduce the following real-valued pressure field $\mathbf{u}^{\ell s} = (u^{\ell s0}, u^{\ell s1}) \in H^1_0(\Omega)^2$, where $u^{\ell s0}$ (resp. $u^{\ell s1}$) represents the real part (resp. the imaginary part) of $\hat{u}^{\ell s}$. A similar operation is carried out with $\hat{f}^{\ell s}$ and $\hat{u}^{\ell s}_{\text{obs}}$ In this specific example, we further assume the mass density constant with

$\rho(\boldsymbol{x}) = \rho_0$ and we only aim at inverting for $\kappa(\boldsymbol{x})$ within a certain subdomain $\Omega^{\mathrm{I}} \subset \mathbb{R}^2$. To do so, one considers some pointwise receptors $\{\boldsymbol{x}_r\}$ positioned around the ROI and one quantifies the misfit between the observed field and a *simulated* pressure field $\hat{u}^{\ell s}$, for each source and frequency with the following cost functional:

$$\mathcal{J}(u,p) = \sum_{\ell,s,j} \int_\Omega (J \circ [u^{\ell sj} - u^{\ell sj}_{\mathrm{obs}}])(\boldsymbol{x}) R(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}. \tag{22}$$

In most applications, the cost functional is simply the sum of the squared difference between u_{obs} and u at each receptor. In such case, we have:

$$\begin{cases} J(\nu) = \nu^2, \\ R(\boldsymbol{x}) = \displaystyle\sum_r \delta(\boldsymbol{x} - \boldsymbol{x}_r). \end{cases}$$

Note that, in our application, the cost functional depends upon u and not on p. However, in some application, some regularization parameter can be added in order to penalize certain solutions. In such case, the cost functional may depends of p as well. Moreover, instead of working in an infinite medium, the computational domain is truncated using Perfectly Matched Layers (PMLs). We thus work in a finite domain $\Omega = \Omega^{\mathrm{PML}} \cup \Omega^{\mathrm{I}}$, with $\Omega^{\mathrm{PML}} \cap \Omega^{\mathrm{I}} = \emptyset$ and impose homogeneous Dirichlet conditions on $\partial\Omega$. Normally, with such a boundary condition, a wave reaching $\partial\Omega$ would be reflected. However a change of variable is performed within Ω^{PML} ensuring that a wave going through Ω^{PML} will not be reflected on its outer boundary. In such a framework, our *complex-valued* model equation can be written as such:

$$\begin{aligned} -\tilde{\nabla} \cdot \left(\frac{1}{\rho(\boldsymbol{x})} \tilde{\nabla} \hat{u}^{\ell s}(\boldsymbol{x}) \right) - \omega_\ell^2 (h \circ p)(\boldsymbol{x}) \hat{u}^{\ell s}(\boldsymbol{x}) &= \hat{f}^{\ell s}(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Omega \\ \hat{u}^{\ell s}(\boldsymbol{x}) &= 0 \qquad \forall \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{23}$$

where $\tilde{\nabla}$ is a modified differential operator that arises from the change of variable that occurs within the PML, i.e.

$$\tilde{\nabla} = \left(\begin{array}{cc} \dfrac{1}{\gamma_x}\dfrac{\partial}{\partial x} & , \quad \dfrac{1}{\gamma_y}\dfrac{\partial}{\partial y} \end{array} \right)^T.$$

Note that, within the PML the terms γ_x and γ_y are complex-valued functions, whereas outside of the PML one has $\gamma_x = \gamma_y = 1$ and $\tilde{\nabla}$ reduces to the standard differential operator ∇. The reader may refer to Bermúdez *et al* for more details on these functions [27]. Rewriting (23) as a coupled PDE whose variables is $\mathbf{u}^{\ell s} = (u^{\ell s 0}, u^{\ell s 1}) \in H_0^1(\Omega)^2$ and integrating against the test function $\mathbf{v}^{\ell s} = (v^{\ell s 0}, v^{\ell s 1}) \in H_0^1(\Omega)^2$, we get the model equation for (u,p) as the following weak form:

$$\langle \mathcal{M}(u,p) - f, v \rangle_{\mathbb{V}^*,\mathbb{V}} = \sum_{\ell,s} \int_\Omega \left\{ \frac{1}{\rho_0} (\nabla \mathbf{u}^{\ell s})^T \mathbf{A} \nabla \mathbf{v}^{\ell s} - \omega_\ell^2 (h \circ p) \mathbf{u}^{\ell s} \cdot \mathbf{v}^{\ell s} - \mathbf{f}^{\ell s} \cdot \mathbf{v}^{\ell s} \right\} \mathrm{d}\boldsymbol{x}$$
$$\forall v = \{\mathbf{v}^{\ell s}\}_{\ell,m} \in \mathbb{V}, \tag{24}$$

with $p \in \mathbb{P}$ being the set of targeted model parameters and $h : \mathbb{R} \to \mathbb{R}^+$ ensuring that, regardless of the values of $p(\boldsymbol{x})$, the reconstructed bulk modulus $\kappa(x)$ will be positive in $\mathbb{R}^2$. Note also that, due to the chosen discretization, $p(\boldsymbol{x})$ is ensured to be bounded. Note that, outside of the PML, $\mathrm{Re}[\frac{1}{\gamma_x}] = \mathrm{Re}[\frac{1}{\gamma_y}] = 1$ and $\mathrm{Im}[\frac{1}{\gamma_x}] = \mathrm{Im}[\frac{1}{\gamma_y}] = 0$. Thus, outside of the PML, $\mathbf{A}$ is equal to the identity matrix, and within the PML, $\mathbf{A}$ is a two by two block matrix that represents the above mentioned change of variable. Note that, in practice, solving (24) may prove to be inefficient because of the coupling between $u^{\ell s0}$ and $u^{\ell s1}$. We thus solve (23) by using the finite element method *with complex variables*. This is only a computational trick and it is not relevant to the behavior of the reconstruction algorithm. Within the proposed framework, at a given iterate k, the quadratic functional $Q_k : \mathbb{U} \times \mathbb{P} \to \mathbb{R}$ can be written as such:

$$Q_k(du_t, dp) = \sum_{\ell,s,j} \int_\Omega \left\{ \left(\left(J' \circ u_k^{\ell sj} \right) du_t^{\ell sj} + \left(J'' \circ u_k^{\ell sj} \right) du_t^{\ell sj} \left[du_n^{\ell sj} + \frac{1}{2} du_t^{\ell sj} \right] \right) R \right.$$
$$\left. -\omega_\ell^2 \left((h' \circ p_k) dp \left[du_t^{\ell sj} + du_n^{\ell sj} \right] + \frac{1}{2} (h'' \circ p_k) dp^2 \; u_k^{\ell sj} \right) \lambda_k^{\ell sj} \right\} \mathrm{d}\boldsymbol{x}.$$

Note that, in the proposed framework, $D_p J(u, p) = 0$ for all $(u, p) \in \mathbb{X}$. Thus, for the sake of concision, some simplifications were made.

4.2. Steps of the proposed method

We now proceed with a description of the key points of the computation of the tangential step dx_t and the quasi-normal step dx_n in the proposed method. We assume here that (u_k, p_k, λ_k) is given .

4.2.1. Computing the quasi-normal step. As previously stated, by setting $dx_n = (du_n, 0)$, one can compute a decent enough solution to (7) by solving the following direct equation:

$$\int_\Omega \left\{ \frac{1}{\rho_0} (\nabla [\mathbf{u}_k^{\ell s} + \mathbf{du}_n^{\ell s}])^T \mathbf{A} \nabla \mathbf{v}^{\ell s} - \omega_\ell^2 (h \circ p_k) [\mathbf{u}_k^{\ell s} + \mathbf{du}_n^{\ell s}] \cdot \mathbf{v}^{\ell s} \right\} \mathrm{d}\boldsymbol{x} = \int_\Omega \mathbf{f}^{\ell s} \cdot \mathbf{v}^{\ell s} \mathrm{d}\boldsymbol{x}$$
$$\forall \mathbf{v}^{\ell s} \in H_0^1(\Omega)^2 \quad \forall \ell, s.$$

The solution $du_n = \{du_n^{\ell sj}\}_{\ell,s,j}$ is then scaled such that (7b) is satisfied. More precisely, each $du_n^{\ell m}$ is scaled such that, globally, $\|du_n\| \leq \zeta \Delta$.

4.2.2. Computing the tangential step. As mentioned above, we can compute the tangential step by solving (13) with the GLTR method. For this purpose, we need to compute the gradient g_k of the reduced cost functional $\hat{Q}_k$ and the product $H_k d$ between a direction $d \in \mathbb{P}$ and the Hessian of the former. From (17), we find that the term g_k of the reduced functional $\hat{Q}_k$ can be written as the sum between the following

two terms:

$$\begin{cases} \nabla_{dp} Q_k(0,0) = -\sum_{\ell,s,j} \omega_\ell^2 (h' \circ p_k) du_n^{\ell s j} \lambda_k^{\ell s j}, \\ \nabla_p \left[\mathcal{M}(u_k, p_k) S_{\text{adj}}(0) \right] = -\sum_{\ell,s,j} \omega_\ell^2 (h' \circ p_k) u_k^{\ell s j} S_{\text{adj}}(0)^{\ell s j}. \end{cases}$$

The term, g_k can thus be written as such:

$$g_k = -\sum_{\ell,s,j} \omega_\ell^2 (h' \circ p_k) \left(u_k^{\ell s j} S_{\text{adj}}(0)^{\ell s j} + du_n^{\ell s j} \lambda_k^{\ell s j} \right), \tag{25}$$

where, for all ℓ, s, the term $\boldsymbol{S}_{\text{adj}}^{\ell s}(0) = (S_{\text{adj}}^{\ell s 0}(0), S_{\text{adj}}^{\ell s 1}(0)) \in H_0^1(\Omega)^2$ is the solution of the following adjoint equation:

$$\int_\Omega \left\{ \frac{1}{\rho_0} (\nabla \boldsymbol{S}_{\text{adj}}^{\ell s}(0))^T \mathbf{A}^T \nabla \tilde{\mathbf{u}}^{\ell s} - \omega_\ell^2 (h \circ p_k) \boldsymbol{S}_{\text{adj}}^{\ell s}(0) \cdot \tilde{\mathbf{u}}^{\ell s} \right\} \mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f}_{\text{adj}}^{\ell s} \cdot \tilde{\mathbf{u}}^{\ell s} \mathrm{d}\boldsymbol{x} \quad \forall \tilde{\mathbf{u}}^{\ell s} \in H_0^1(\Omega)^2 \quad \forall \ell, s, \tag{26}$$

where, for all ℓ, s, one has $\boldsymbol{f}_{\text{adj}}^{\ell s} = (f_{\text{adj}}^{\ell s 0}, f_{\text{adj}}^{\ell s 1}) \in L^2(\Omega)^2$ and this source term is derived from both the gradient and the Hessian matrix of the cost functional (22) as:

$$f_{\text{adj}}^{\ell s j} = -\left[(J' \circ u_k^{\ell s j}) + (J'' \circ u_k^{\ell s j}) du_n^{\ell s j} \right] R.$$

Note that, in practice, similarly to the direct equation, instead of solving (26), we solve the equation adjoint to (23) by using the finite element method *with complex variables* and with $\hat{f}_{\text{adj}} = (f_{\text{adj}}^{\ell s 0} + i f_{\text{adj}}^{\ell s 1})$ as the source term. Similarly, from (18), we find that the product between the Hessian operator and a direction $d \in \mathbb{P}$ can be written as the following three terms:

$$\begin{cases} \nabla_p \left[\mathcal{M}(u_k, p_k) \mu_{\text{adj}} \right] = -\sum_{\ell,s,j} \omega_\ell^2 (h' \circ p_k) u_k^{\ell m n} \mu_{\text{adj}}^{\ell s j}, \\ \nabla^2_{dp\, du_t} Q_k(du_t, dp) \mu = -\sum_{\ell,s,j} \omega_\ell^2 (h' \circ p_k) \mu^{\ell s j} \lambda_k^{\ell s j}, \\ \nabla^2_{dp\, dp} Q_k(du_t, dp) d = -\sum_{\ell,s,j} \omega_\ell^2 u_k^{\ell s j} \lambda_k^{\ell s j} (h'' \circ p_k) d. \end{cases}$$

The quadratic term, H_k can thus be written as such:

$$H_k d = -\sum_{\ell,s,j} \omega_\ell^2 (h' \circ p_k) \left(u_k^{\ell s j} \mu_{\text{adj}}^{\ell s j} + \mu^{\ell s j} \lambda_k^{\ell s j} \right) + \omega_\ell^2 u_k^{\ell s j} \lambda_k^{\ell s j} (h'' \circ p_k) d, \tag{27}$$

where $\mu^{\ell s} \in H_0^1(\Omega)^2$ is, for all ℓ, s, the solution of the following direct equation:

$$\int_\Omega \left\{ \frac{1}{\rho_0} (\nabla \boldsymbol{\mu}^{\ell s})^T \mathbf{A} \nabla \mathbf{v}^{\ell s} - \omega_\ell^2 (h \circ p_k) \boldsymbol{\mu}^{\ell s} \cdot \mathbf{v}^{\ell s} \right\} \mathrm{d}\boldsymbol{x} = \int_\Omega \omega_\ell^2 (h' \circ p_k) d\ \mathbf{u}_k^{\ell s} \cdot \mathbf{v}^{\ell s} \mathrm{d}\boldsymbol{x} \quad \forall \mathbf{v}^{\ell s} \in H_0^1(\Omega)^2 \quad \forall \ell, s$$

and where $\mu_{\text{adj}}^{\ell s} \in (H_0^1(\Omega))^2$ is, for all ℓ, s, the solution of the following adjoint equation:

$$\int_\Omega \left\{ \frac{1}{\rho_0} (\nabla \boldsymbol{\mu}_{\text{adj}}^{\ell s})^T \mathbf{A}^T \nabla \tilde{\mathbf{u}}^{\ell s} - \omega_\ell^2 (h \circ p_k) \boldsymbol{\mu}_{\text{adj}}^{\ell s} \cdot \tilde{\mathbf{u}}^{\ell s} \right\} \mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f}_{\text{adj2}}^{\ell s} \cdot \tilde{\mathbf{u}}^{\ell s} \mathrm{d}\boldsymbol{x}$$
$$\forall \tilde{\mathbf{u}}^{\ell s} \in H_0^1(\Omega)^2 \quad \forall \ell, s$$

where, for all ℓ, s one has $\boldsymbol{f}_{\text{adj2}}^{\ell s} = (f_{\text{adj2}}^{\ell s 0}, f_{\text{adj2}}^{\ell s 1}) \in L^2(\Omega)^2$ and this source term can be written as such:

$$f_{\text{adj2}}^{\ell s j} = \omega_\ell^2 (h' \circ p_k) d \ \lambda_k^{\ell s j} - (J'' \circ u_k^{\ell s j}) \mu^{\ell s j} R.$$

Note that μ (respectively μ_{adj}) satisfies the wave equation whose source term quantifies how the residual of the model equation (23) (respectively the adjoint equation corresponding to (23)) varies when the coefficients $(h \circ p_k)$ vary along the direction d. Finally, once dp has been found, du_t is computed by applying the solution operator S to dp. In this specific framework, this is done by solving the following direct equation:

$$\int_\Omega \left\{ \frac{1}{\rho_0} (\nabla \mathbf{d}\mathbf{u}_t^{\ell s})^T \mathbf{A} \nabla \mathbf{v}^{\ell s} - \omega_\ell^2 (h \circ p_k) \mathbf{d}\mathbf{u}_t^{\ell s} \cdot \mathbf{v}^{\ell s} \right\} \mathrm{d}\boldsymbol{x} = \int_\Omega \omega_\ell^2 (h' \circ p_k) dp \ \mathbf{u}_k^{\ell s} \cdot \mathbf{v}^{\ell s} \mathrm{d}\boldsymbol{x}$$
$$\forall \mathbf{v}^{\ell s} \in H_0^1(\Omega)^2 \quad \forall \ell, s.$$

Upon solving (13) with the GLTR method, we need to compute the linear term of the reduced cost functional $\hat{Q}_k$ with (25). Then at each iteration of the GLTR algorithm, given a direction $d \in \mathbb{P}$, we need to compute the product with the Hessian of the reduced cost functional with (27).

4.2.3. Updating the Lagrange multiplier. The new lagrange multiplier, $\lambda_{k+1} = S_{\text{adj}}(dp)$, is then computed by solving the following adjoint equation:

$$\int_\Omega \left\{ \frac{1}{\rho_0} (\nabla \boldsymbol{\lambda}_{k+1}^{\ell s})^T \mathbf{A}^T \nabla \tilde{\mathbf{u}}^{\ell s} - \omega_\ell^2 (h \circ p_k) \boldsymbol{\lambda}_{k+1}^{\ell s} \cdot \tilde{\mathbf{u}}^{\ell s} \right\} \mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f}_{\text{mult}}^{\ell s} \cdot \tilde{\mathbf{u}}^{\ell s} \mathrm{d}\boldsymbol{x}$$
$$\forall \tilde{\mathbf{u}}^{\ell s} \in H_0^1(\Omega)^2 \quad \forall \ell, s$$

where, for all ℓ, s one has $\boldsymbol{f}_{\text{mult}}^{\ell s} = (f_{\text{mult}}^{\ell s 0}, f_{\text{mult}}^{\ell s 1}) \in L^2(\Omega)^2$ and this source term is derived from both the gradient and the Hessian operator of the cost functional (22) as:

$$f_{\text{mult}}^{\ell s j} = \omega_\ell^2 (h' \circ p_k) dp \ \lambda_k^{\ell s j} \tilde{u}^{\ell s j} - \left[(J' \circ u_k^{\ell s j}) + (J'' \circ u_k^{\ell s j}) (du_n^{\ell s j} + du_t^{\ell s j}) \right] R.$$

Note that, if $dp = 0$, then $f_{\text{mult}}^{\ell s j} = f_{\text{adj}}^{\ell s j}$ for all ℓ, s.

Remark 3 *TR-SQP algorithms can be globally convergent under certain assumptions (see Dennis et al [14] for the general case and Heinkenschloss et al [10] and Ziems and Ulbrich [11] for the PDE constrained optimization case. However, this requires the solution operators S and S_{adj} to be uniformly bounded, which, to our knowledge, is not the case for the Helmholtz equation. Note that this is an intrinsic feature of the problem under consideration that also applies to other optimization methods.*

4.3. Results and discussion

4.3.1. Forward simulations. We assess in this section the performances of the proposed method on a synthetic example with a configuration notably relevant to medical imaging. Our results are also compared to a standard method that uses the adjoint method to compute the gradient of the reduced functional $\hat{\mathcal{J}}$ and compute a descent direction with the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method. Following Bernard *et al*, who studied several possible configurations in medical imaging [28], we choose to take what they determined to be the optimal configuration for the IWP. The sources and receptors surround the object so as to have a regime of both transmitted and reflected waves. In addition, as suggested in the paper, we have used fewer sources than receivers because the cost of the method depends directly on the number of sources. We used enough receivers to sample the wavefield correctly. We assumed the source term as known and used the latter in the inversions. In experimental cases, full knowledge of the source term may require an additional inversion to properly represent the radiation pattern of the source term and the temporal source function. This can be done as a pre-processing step by making measurements in the reference media (e.g. water). Another possible solution is to define a cost functional independent of the source term (e.g. Bachmann and Tromp [29]) in order to be unaffected by possible artifacts coming from the lack of knowledge of this term. The testing configuration is as follows: we aim at reconstructing a bulk modulus distribution $\kappa(\boldsymbol{x})$ that characterizes an unknown *object*, which is embedded within a homogeneous background domain, thus defining a parameter to optimize $p \in \mathbb{P}$ such that $p(\boldsymbol{x}) = -\ln \kappa(\boldsymbol{x})$ to ensure a positivity constraint. We consider strong contrasts, the sought objects has two components, which have respectively a 50% and 100% contrast compared to the reference medium. Moreover these components are close enough to produce multiple scattering effects. While the mass density is assumed to be uniform within the whole domain, with $\rho_0 = 1$, the targeted bulk modulus is the heterogeneous distribution shown Figure 1-1, with $\kappa(\boldsymbol{x}) = \kappa_0 = 2$ for the homogeneous background medium. The said object is monitored using 8 sources and 128 receptors positioned in circle around it and the ROI, denoted Ω^{I}, is defined as a 2 m by 2 m box. The computational domain Ω itself is divided into three subdomains: Ω_p^{I} where p is inverted, $\Omega_{p_0}^{\mathrm{I}}$ where p is considered to be homogeneous with $p(\boldsymbol{x}) = p_0$ and so that $\Omega^{\mathrm{I}} = \Omega_p^{\mathrm{I}} \cup \Omega_{p_0}^{\mathrm{I}}$, and the PML region Ω^{PML}, which ensures that waves are not reflected once they reach $\partial\Omega$. Figure 1-2 shows the mesh used to solve the direct problem with a different color for each of these subdomains. Synthetic data is generated using the finite element method on a 75×75 grid with P^6 shape functions, while ω_ℓ together with the associated wavelengths (within the background medium) are chosen as in Table 1. Doing so, the wavelength of the probing wave corresponds to, for the first frequency, roughly the diameter of the object and for, the tenth frequency, roughly a tenth of the diameter of the objects.

The sources are driven using a single time-domain signal, which is defined as a Gaussian function multiplied by a sinus function, see Figure 2-1). This master signal is

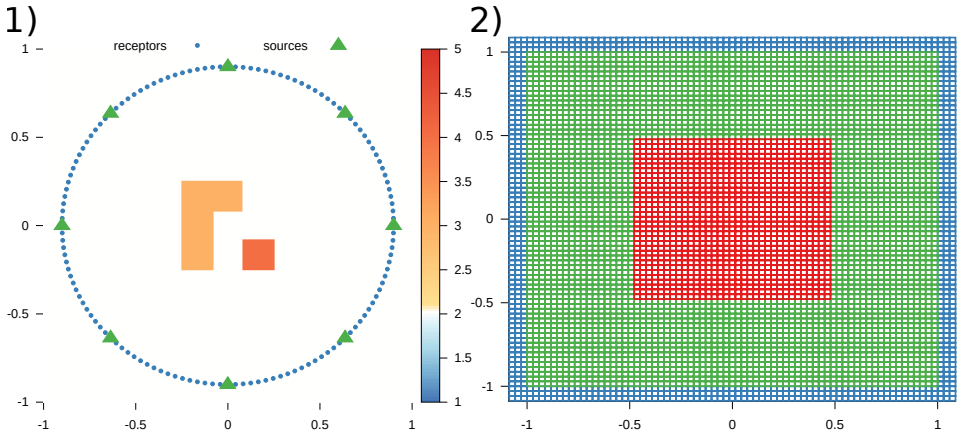

Figure 1: Test case considered 1) The map of parameter κ that characterizes the targeted object. In blue, the 128 receptors disposed around the object that we are trying to recover. In green, the 8 sources disposed around said object. 2) The mesh of our computational domain Ω. In red Ω_p^{I} where p is not homogeneous. In green, $\Omega_{p_0}^{\mathrm{I}}$ where p is homogeneous. In blue Ω^{PML} where we placed the PMLs so that the wave propagated through our medium are not reflected upon reaching the boundaries of our computational domain.

ω_ℓ	14	28	42	56	70	84	99	113	127	141
wavelength	0.63	0.31	0.21	0.16	0.13	0.11	0.09	0.08	0.07	0.06

Table 1: Chosen set $\{\omega_\ell\}$ with associated wavelength in the background medium.

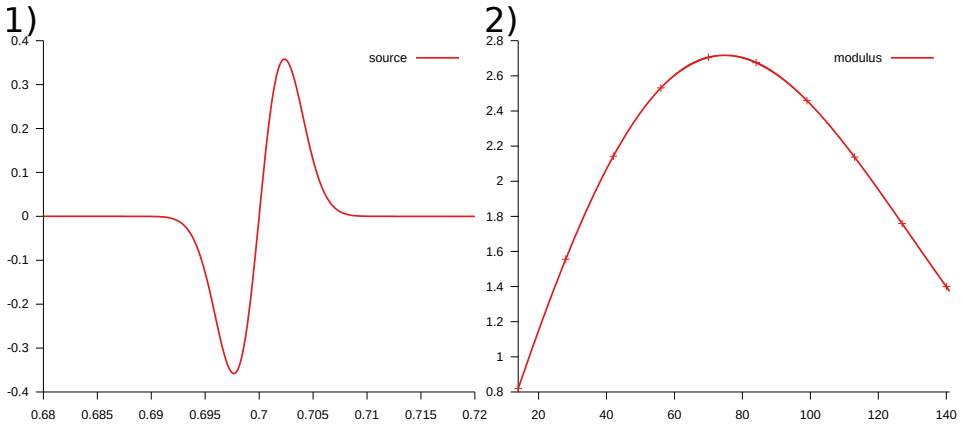

Figure 2: Here we show our source term in both time and frequency domains. 1) The source term in the time domain, displayed, for the sake of clarity from t = 0.68 s to t = 0.72 s. Note that the source was computed from t = 0 s to t = 20 s so that the DFT of the signal is properly sampled. 2) The modulus of the source term in the frequency domain.

computed on a time window that ranges from t = 0 s to t = 20 s, and the time-harmonic

source signals are extracted from its DFT. This master signal is selected so that all of the exciting sources have comparable amplitudes (c.f. Figure 2-2)), thus ensuring that all of the generated data would contribute significantly to the inversion. Figure 3 shows solutions of the forward problem that are computed for some frequencies originating from the master signal. The data set considered is obtained by sampling the solution of the forward problem at each receptor $\boldsymbol{x}_r$, for each frequency ω_ℓ and for each source location $\boldsymbol{x}_s$.

4.3.2. Inversion frequency by frequency starting at low frequency. The inversion proceeds as follows, we solve the inverse problem with the data corresponding to the ℓ^{th} frequency starting from the inversion results of the $(\ell-1)^{th}$ frequency data. The initialization of the inversion associated with the first frequency is homogeneous with $p_0 = -\ln \kappa_0$. The convergence criterion for our method (resp. the L-BFGS) is $\|\nabla \mathcal{L}(u_k, p_k, \lambda_k)\|_{\mathbb{X}\times\mathbb{V}} \leq \epsilon$ (resp. $\|\nabla \hat{\mathcal{J}}(p)\|_{\mathbb{P}} \leq \epsilon$) with $\epsilon = 10^{-3}$, see Algorithm 1, which we will now refer to as residual for the sake of concision. Our method was allowed to perform 100 iterations of the GLTR algorithm while the L-BFGS was allowed to maintain a memory of 100 updates for a fair comparison. Moreover, if we could not find a satisfying descent direction with the L-BFGS algorithm, we would use the gradient of the reduced cost functional as a descent direction.

We first compare the two methods by performing an inversion frequency by frequency starting from $\omega_0 = 14$ and with noise-free data. We show Figure 4 the results yielded by both methods, frequency by frequency. We can see that the low frequency data lead to blurred bulk modulus distributions and that adding higher frequencies allows us to recover more detailed and sharper maps. We can observe that both methods yield comparable results qualitatively. We also provide, Figure 5, some cross-sections for these results. We can observe that, for the lowest frequencies, the two method seem to find different results: For ω_0, our solution exhibits less oscillations around the correct targeted solution, see Fig. 5a), whereas it oscillates more around the correct targeted solution at ω_2, see Fig. 5b). However, for higher frequencies, the two methods yield similar results, see the figures 5c-d). We can see that the solutions yielded by both methods seem to oscillate around the targeted bulk modulus distribution. It seems that the cost functional rewards, or at least does not penalize, such solutions. Note that, if

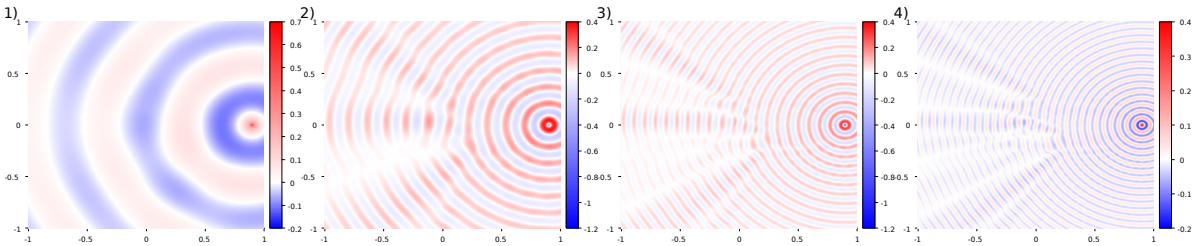

Figure 3: 1-4) The real part of the solution of the direct problem for $\{\omega_\ell\} = \{14, 56, 99, 141\}$.

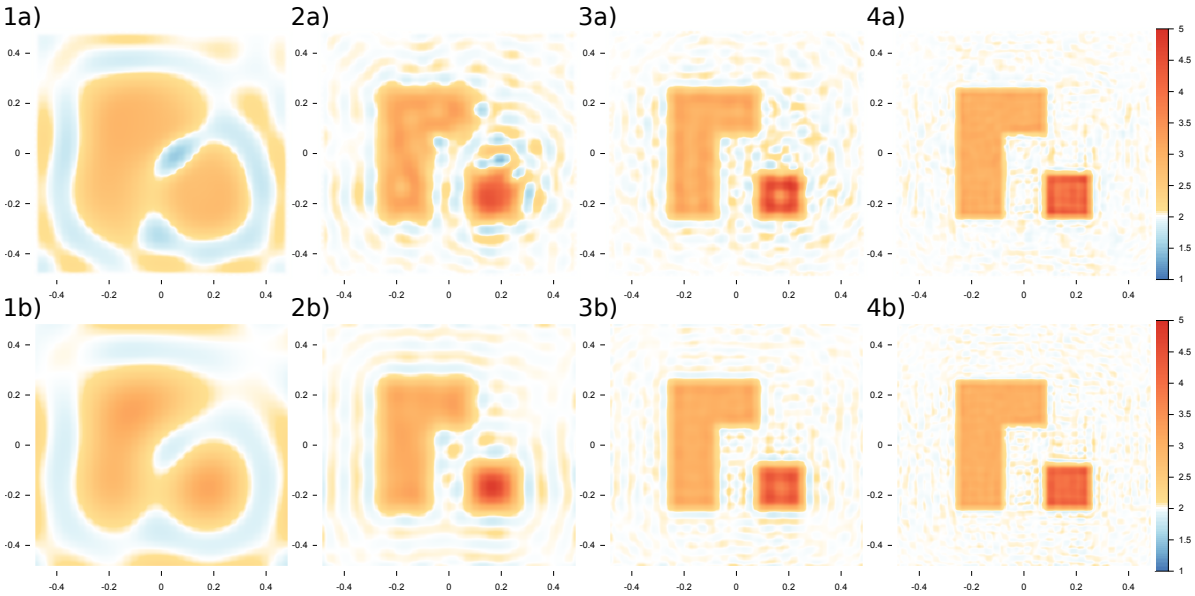

Figure 4: Results of inversion frequency by frequency. We present figure 1)-4) the results yielded by both our method (top panels a) and the L-BFGS method (bottom panels b). For the sake of concision, we only show the results of the inversion for $\{\omega_\ell\} = \{14, 42, 84, 141\}$, respectively.

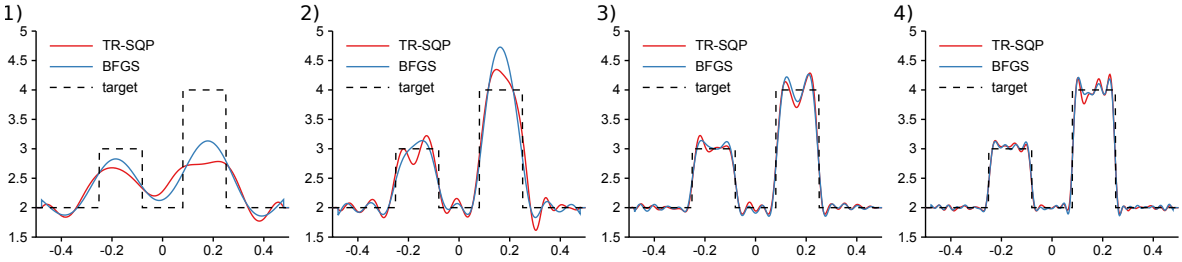

Figure 5: Cross-sections of the results of inversion frequency by frequency. figure 1)-4) the results yielded by both our method (red) and the L-BFGS method (blue). For the sake of concision, we only show the results of the inversion for $\{\omega_\ell\} = \{14, 42, 84, 141\}$, respectively.

needed, this can be remedied by adding a Total Variation (TV) regularization term to the cost functional.

We should mention that both method converged in roughly 3 hours. Moreover, TR-SQP was 6.6% slower than L-BFGS. However, the L-BFGS method converged after a total of 168 iterations, the TR-SQP method converged after a total of 33 iterations. A TR-SQP iteration is thus roughly 5 times more expensive than a L-BFGS iteration. As for the memory consumption, the L-BFGS consumes 9.5Gb of RAM and the TR-SQP consumes 13Gb of RAM. We should mention that the vectors y_k, s_k and q_k generated by both the L-BFGS method and the GLTR method are stored on the hard drive.

4.3.3. Inversion with noisy data. Here, we compare the two methods by performing inversions from noisy data, still frequency by frequency and starting from $\omega_0 = 14$. With a uniformly distributed white noise added to the data of amplitude 10%. We show Figure 6 and Figure 7 the results yielded by both methods, frequency by frequency.

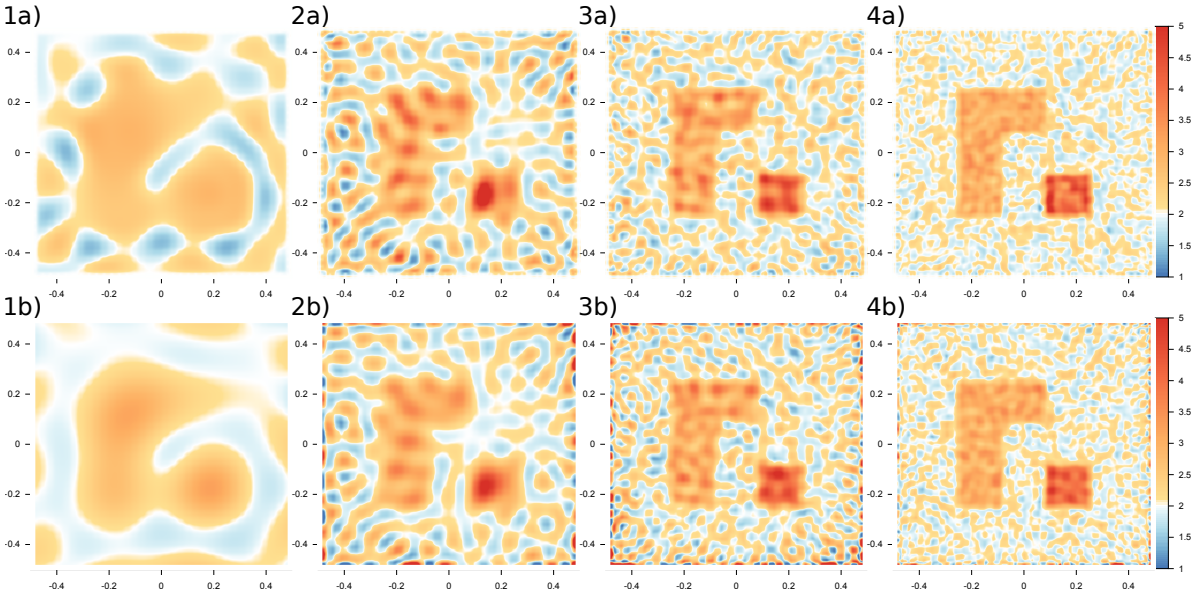

Figure 6: Results of inversion frequency by frequency with 10% of uniformly distributed white noise added to the data. We present figure 1)-4) the results yielded by both our method (top panels a) and the L-BFGS method (bottom panels b). For the sake of concision, we only show the results of the inversion for $\{\omega_\ell\} = \{14, 42, 84, 141\}$, respectively.

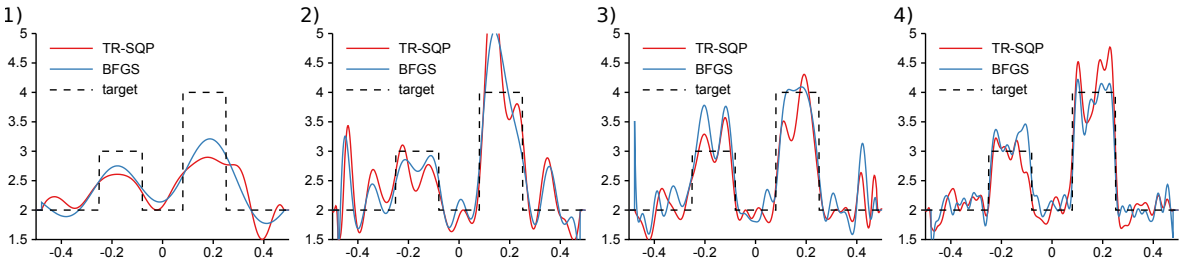

Figure 7: Cross-sections of the results of inversion frequency by frequency with 10% of uniformly distributed white noise added to the data. figure 1)-4) the results yielded by both our method (red) and the L-BFGS method (blue). For the sake of concision, we only show the results of the inversion for $\{\omega_\ell\} = \{14, 42, 84, 141\}$, respectively.

The images obtained when inverted with noise show spurious oscillations both inside the reconstructed objects and in the area surrounding them. These artifacts are probably due to a kind of overfitting of the noisy data because we do not use any mechanism to take this noise into account in the cost function. Table 2 shows that both methods

converge towards solutions that have comparable cost values so we cannot discriminate between the results of the two methods. It seems that these oscillations are somewhat

ω	TR-SQP			L-BFGS	
	#it	$\mathcal{J}(u)$	$\hat{\mathcal{J}}(p)$	#it	$\hat{\mathcal{J}}(p)$
14	3	1.36e-02	1.39e-02	9	1.37e-02
28	6	2.77e-02	2.81e-02	18	2.83e-02
42	11	3.08e-02	3.09e-02	65	3.08e-02
56	13	2.82e-02	2.82e-02	39	2.93e-02
70	12	1.85e-02	1.85e-02	108	1.80e-02
84	19	1.50e-02	1.50e-02	142	1.45e-02
99	11	7.39e-03	7.39e-03	91	6.98e-03
113	14	3.13e-03	3.13e-03	47	3.32e-03
127	13	1.62e-03	1.62e-03	50	1.61e-03
141	9	6.23e-04	6.24e-04	43	6.09e-04

Table 2: Comparison between our method (TR-SQP) and the L-BFGS method with 10% of uniformly distributed white noise and starting from $\omega_0 = 14$. #it corresponds to the number of iterations that were required to meet the convergence criterion. $\mathcal{J}(u)$ corresponds to the value taken by the cost functional (c.f. Eq. (22)) and $\hat{\mathcal{J}}(p)$ corresponds to the value taken by the reduced cost functional (c.f. Eq. (1)).

more pronounced with the TR-SQP method than with the L-BFGS. But the latter shows strong artifacts on the edge of the domain which is poorly resolved because situated outside the network of sources and receivers. The TR-SQP method does not show this type of artifacts on the edge of the domain, we believe that this is due to an intrinsic regularization mechanism of the TR method.

To mitigate the phenomenon of overfitting we can either regularize by adding a penalty on the model in the cost function which would penalize the too strong oscillations (TV regularization for example) or use all the frequencies simultaneously in a single inversion. This leads a kind of regularisation effect because the noise we used is decorrelated between the frequencies. We thus compare both methods by performing an inversion with all frequencies simultaneously, see Figure 8. We can see that both methods give similar results qualitatively, but we can compare quantitatively on Figure 9 the values taken by the cost functional and the residual of the model equation as functions of the number of iterations. We can see that our method finds a bulk modulus distribution that corresponds to a lower value of the cost functional, and this is achieved with less iterations than the standard L-BFGS method. We should mention that, for the sake readability, the curve does not show all the L-BFGS iterations. However, in the end the L-BFGS method produces a result that corresponds to a value of 8.62e-04 while the TR-SQP produces a result that corresponds to a value of 7.85e-05. We can also see, in Figure 9, that between iterations 5 and 10, the model error starts to increase. After that, the model residual starts to decrease. Similarly, between iteration

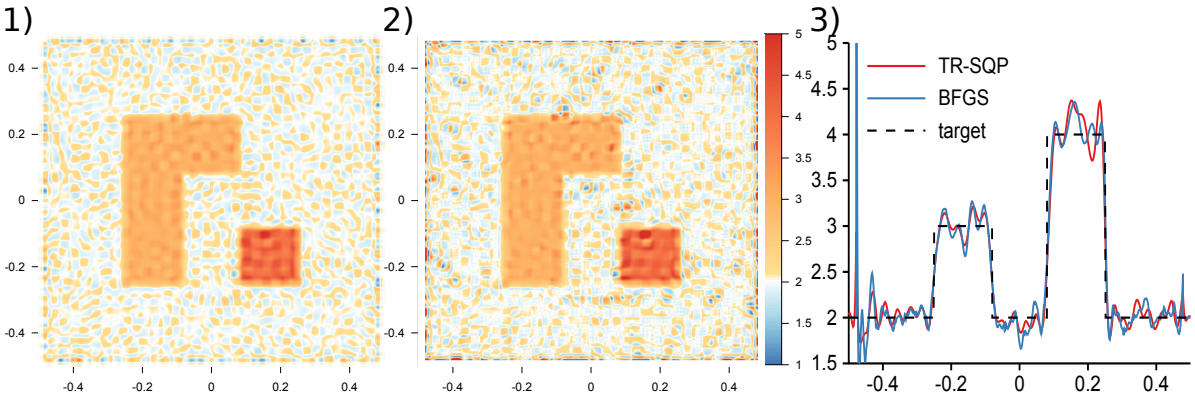

Figure 8: Results of inversion with all frequencies simultaneously with 10% of uniformly distributed noise added to the data. 1) The result yielded by our method. 2) The result yielded by the L-BFGS method. 3) Cross-sections of the results yielded by method (red) and by the L-BFGS method (blue).

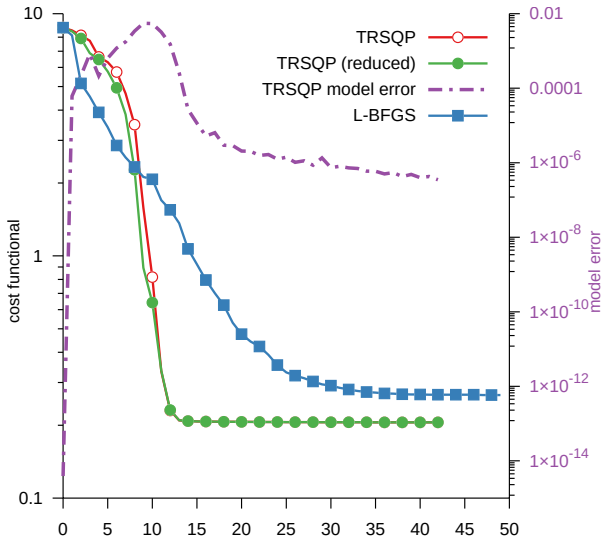

Figure 9: Results of inversion with all frequencies simultaneously. On the left axis, the values taken by the cost functional vs the number of iterations with both methods. In red, the value taken by the cost functional $\mathcal{J}(u)$ (c.f. Eq. (22)) with the TR-SQP method, in green, the value taken by the reduced cost functional $\hat{\mathcal{J}}(p)$ (c.f. Eq. (1)) with the TR-SQP method and in blue, the value taken by the reduced cost functional $\hat{J}(p)$ with the L-BFGS method. On the right axis, in dashed purple, the model error commited by our method vs the number of iterations.

5 and 10, the cost functional and the reduced cost functional exhibit the most drastic difference. Furthermore, we can see that between iteration 8 and 9, when the model residual is at its peak, our method finds a step that is much better according to the cost functional than to the reduced cost functional. This leads us to believe that, by not strictly enforcing the model equation at each iteration, our method is able to find steps that are good according to the cost functional and not so good according to the reduced

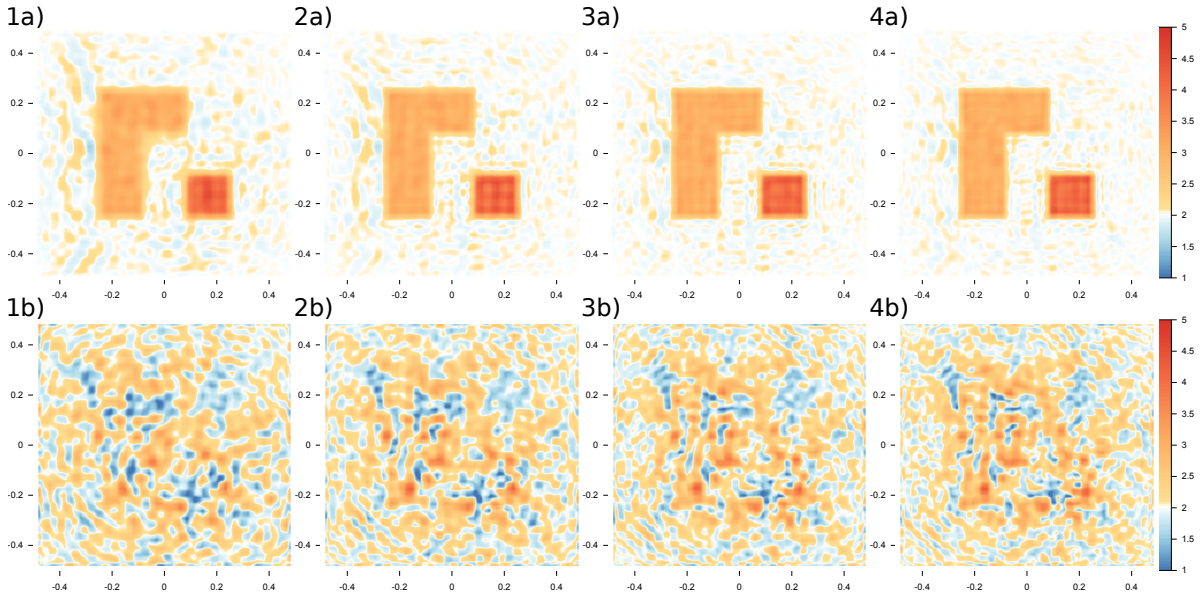

Figure 10: Results of inversion frequency by frequency starting from high frequencies. We present figure 1)-4) the results yielded by both our method (top panels a) and the L-BFGS method (bottom panels b) with $\{\omega_\ell\} = \{99, 113, 127, 141\}$, respectively.

cost functional. This is interesting because, it is believed that, not strictly enforcing the model equation at each step can lead to faster convergence and even, to avoiding certain local minima [23].

4.3.4. Inversion frequency by frequency with high frequencies only. In practice, the data corresponding to the lowest frequencies are generally not available. Therefore, we finally compare the two methods by performing some inversions on noise-free data and frequency by frequency, but starting directly from a high frequency, i.e. $\omega_6 = 99$, at which our method yielded qualitatively good results without noise. We show Figure 10, the results yielded by both methods, frequency by frequency. We can observe that, thanks to the TR constraint, our method finds qualitatively better solution than the L-BFGS method. Moreover, according to the values reported in Table 3, the lowest frequencies solutions are quantitatively better according to both the cost functional and the reduced cost functional. However, this is not true for higher frequencies. This means that 1) the L-BFGS method was stuck into a local minimum and 2) the cost functional is not able to discriminate between the solutions proposed by both methods. The failure of the L-BFGS method is due to the cycle skipping effect [28]. When the phase match between the field computed in the initial model compare to the data is more than half a wavelength, the method cannot reconstruct the object. This is why, generally, the strategy of starting at the lowest frequencies and resetting the method at each new frequency starting from the model obtained at the previous frequency is used. In some experimental configurations sufficiently low frequencies may be missing. This can be a problem of instrumentation but also the physical characteristics of the

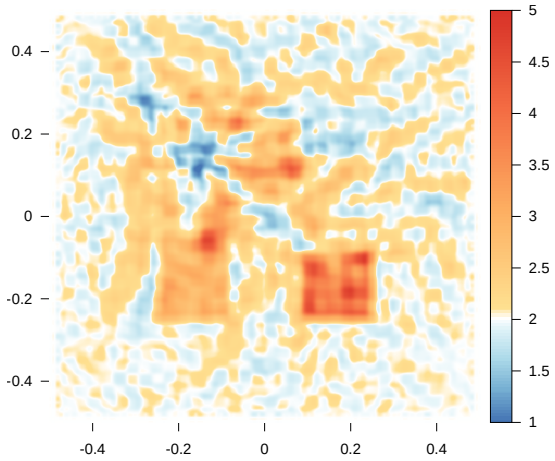

Figure 11: Final result of inversion frequency by frequency starting from $\omega_7 = 113$ yielded by our method. We can see that the bottom object is correctly retrieved, while a piece of the larger object is missing.

object themselves (in the case of strong contrasts) can lead cycle skipping. A solution would be to have an initial model closer to the real model to avoid this effect. But again this is not always possible in real cases. The advantage of the method proposed here is to be less sensitive to the cycle skipping, as shown in the result of Figure 3. The success of our method is partly due to our TR constraint. Indeed, during the first few iterations, because of the cycle skipping effect, the gradient of the reduced cost functional $\hat{Q}_k$ tends to show the boundary of the object of interest, but not the interior. However, such a descent direction tends to be poor according to our penalty function (c.f. Eq. B.1). See section 2.3 and Appendix B for more details. The TR radius is thus decreased and our method is forced to find a *smoother dp*, which in turns allows our method to recover a very blurred object. After a few iterations like this, the gradient of $\hat{Q}_k$ starts to show both the boundary and the interior of the object, as if we had a good first guess. Nevertheless, starting at the frequency $\omega_7 = 113$ our method also shows difficulties to converge, the smaller object is correctly found and only a part of the large object is recovered (c.f. Figure 11). This leads us to believe that our method is still sensitive to the cycle skipping effect to some extents. In other words, while our method is guaranteed to converge to *a* solution of (3), it is not guaranteed to converge towards the targeted *true* parameters. We should recall that the FWI is not a convex optimization problem hence there no mathematical proof that any method will converge towards a global minimum nor that said global minimum is unique. Hence the only way to truly prevent the cycle-skipping effect from happening would be to reformulate the FWI as a convex optimization problem, which remains an open question.

ω	TR-SQP			L-BFGS	
	#it	$\mathcal{J}(u)$	$\hat{\mathcal{J}}(p)$	#it	$\hat{\mathcal{J}}(p)$
99	22	2.74e-05	2.78e-05	59	2.66e-04
113	5	1.13e-05	1.16e-05	50	1.34e-04
127	5	5.77e-06	5.81e-06	42	9.69e-05
141	3	2.10e-04	2.06e-04	39	4.50e-05

Table 3: Comparison between our method (TR-SQP) and the L-BFGS method on a noise-free example starting from $\omega_6 = 99$. #it corresponds to the number of iterations that were required to meet the convergence criterion. $\mathcal{J}(u)$ corresponds to the value taken by the cost functional (c.f. Eq. (22)) and $\hat{\mathcal{J}}(p)$ corresponds to the value taken by the cost functional (c.f. Eq. (1)).

5. Conclusion

In this work we have presented a, to our knowledge, new TR-SQP method. We proposed a new TR constraint that acts as a TV regularization on each sub-problems. We also applied, as far as we know, for the first time the *second-order adjoint method* to the QP subproblems. This allowed us to propose a new way to solve each subproblems with a Krylov method, the GLTR. Our method was tested on a configuration relevant to medical imaging. Our method works in an extended search space and was able to find, the same results as the standard L-BFGS with a reduced search space. Because of the TR constraint, our method seems sensitive to noise in a different way than the standard approach. More precisely, while the standard approach generates solutions with spikes that are smaller than the wavelength of the lowest frequency treated, such solutions are explicitly penalized by our TR constraint. Moreover, we have seen that, contrary to the standard approach, our method is able to correctly identify the model parameters even when low frequencies are not available. However, we believe that such a result is due to our special TR constraint and not to the extended search space approach. Both methods have similar computational costs. More precisely, an iteration from our method is much more expensive than a L-BFGS iteration but it takes significantly less iterations for our method to reach global convergence. It is possible to lower the cost of a TR-SQP iteration by using a inexpensive approximation of the Hessian when solving the tangential subproblem. This could be done by using a standard quasi-Newton method or by removing the terms of the Hessian that require to solve PDEs. However, this may lead to lower convergence speeds. Alternatively, we can improve the conditioning of each subproblems by changing the TR constraint. More precisely, we could choose a TR constraint that involves a linear operator $B_k : \mathbb{X} \to \mathbb{X}$ that "is similar" to the inverse of the Hessian of the cost functional of each sub-problems. Although, it is possible that, with a different TR our method would not perform as well when low frequencies are missing.

Further investigations include, expanding the method to the time domain, which is

challenging because of the memory requirement of our method in the time domain, as noted by Epanomeritakis *et al* [37], using our TR constraint on a reduced space method in order to see if we can correctly reconstruct the model parameters without the low frequencies, using a TR constraint inspired by the work of [31, 30], seeking good and inexpensive approximation of the Hessian and applying our method to, among others, geophysical problems.

Appendix A. The adjoint method

Suppose we want to solve the following PDE constrained optimization problem:

$$\underset{(du,\,dp)\,\in\,\mathbb{U}\times\mathbb{P}}{\text{minimize}} \qquad Q_k(du, dp) \tag{A.1a}$$

$$\text{subject to} \qquad M_k du = F_k dp, \tag{A.1b}$$

This problem can be solved by finding a triplet $(du^*, dp^*, \lambda^*) \in \mathbb{U} \times \mathbb{P} \times \mathbb{V}$ where $\mathbb{U}$, $\mathbb{P}$, $\mathbb{V}$ are Hilbert spaces, such that

$$\nabla \mathcal{L}(du^*, dp^*, \lambda^*) = 0, \tag{A.2}$$

where $\mathcal{L}$ is the *Lagrange function* corresponding to (A.1), i.e.

$$\mathcal{L}(du, dp, \lambda) = Q_k(du, dp) + \langle M_k du - F_k dp, \lambda \rangle_{\mathbb{V}^*,\mathbb{V}}.$$

Equation (A.2) are often refereed to as the first order KKT conditions associated with (A.1). However, this approach requires to solve a large system of equations, which can be prohibitively expensive. Another approach, consist in enforcing the model constraint (A.1b) by introducing a *solution operator* $S : \mathbb{P} \to \mathbb{U}$ such that:

$$M_k S(dp) = F_k dp \qquad \forall dp \in \mathbb{P}.$$

By using this operator, (A.1) can be rewritten as an unconstrained optimization problem where one tries to find the minimum of the reduced cost functional $\hat{Q}_k(dp)$ defined as:

$$\hat{Q}_k(dp) := Q_k(S(dp), dp).$$

Instead of directly computing the differential of $\hat{Q}_k$, which can be a convoluted task, especially for the second-order derivatives, it is possible to use the *adjoint method* (*resp.* the *second-order adjoint method*) that allows to compute the gradient (*resp.* the product between a direction and the Hessian operator) of the reduced cost functional $\hat{Q}_k$ without computing the derivatives of the solution operator.

Appendix A.1. Computing the gradient

The keystone of the adjoint method is that the reduced cost functional $\hat{Q}(dp)$ satisfies the following identity:

$$\hat{Q}_k(dp) = \mathcal{L}(S(dp), dp, \lambda) \qquad \forall \lambda \in \mathbb{V}.$$

Thus the Gateaux derivative of $\hat{Q}$ can be written as such:

$$D\hat{Q}_k(dp) = D_{S(dp)}\mathcal{L}(S(dp), dp, \lambda) \circ DS(dp) + D_{dp}\mathcal{L}(S(dp), dp, \lambda) \qquad \forall \lambda \in \mathbb{V}.$$

Moreover, choosing λ_{adj} such that:

$$M_k^* \lambda_{\text{adj}} = -D_u Q_k(u, dp) \tag{A.3}$$

we ensure that

$$\left\langle D_{du}\mathcal{L}(du, dp, \lambda_{\text{adj}}), \tilde{du} \right\rangle_{\mathbb{U}^*,\mathbb{U}} = 0 \qquad \forall \tilde{du} \in \mathbb{U}.$$

and thus $D\hat{Q}_k(dp) = D_{dp}\mathcal{L}(S(dp), dp, \lambda_{\text{adj}})$. Equation (A.3) is often refered to as the *adjoint equation* for it involves the *adjoint model* M_k^*. Finally, this allows us to write the differential of the reduced cost functional $\hat{Q}_k$ as:

$$D\hat{Q}_k(dp) = D_{dp}Q_k(S(dp), dp) - F_k^*\lambda_{\text{adj}}.$$

Note that, according to (A.3), λ_{adj} depends on dp. This is not terribly important as for now, but it will be when computing the product between a direction and the Hessian. We thus introduce another solution operator $S_{\text{adj}} : \mathbb{P} \to \mathbb{V}$, which satisfies the following identity:

$$M_k^* S_{\text{adj}}(dp) = -D_{S(dp)}Q_k(S(dp), dp).$$

The term λ_{adj} can thus be expressed as $\lambda_{\text{adj}} = S_{\text{adj}}(dp)$.

Appendix A.2. Computing the product between a direction and the Hessian

Following [22, 21], we introduce a function $G : \mathbb{U} \times \mathbb{P} \times \mathbb{V} \to \mathbb{P}^*$ and a reduced function $\hat{G} : \mathbb{P} \to \mathbb{P}^*$ defined as such:

$$\begin{aligned} G(du, dp, \lambda) &= D_{dp}Q_k(du, dp) - F_k^*\lambda, \\ \hat{G}(dp) &= G(S(dp), dp, S_{\text{adj}}(dp)). \end{aligned}$$

The second-order adjoint method starts by defining two new functionals: Φ_d, given $d \in \mathbb{P}$, which we aim at differentiating and that is defined as:

$$\Phi_d(dp) = \left\langle \hat{G}(dp), d \right\rangle_{\mathbb{P}^*,\mathbb{P}},$$

and a Lagrange functional $\mathcal{L}_2$ that reads:

$$\begin{aligned} \mathcal{L}_2(du, dp, \lambda, g, \mu_{\text{adj}}, \mu, \mu_{\text{dir}}) = \langle M_k du - F_k dp, \mu_{\text{adj}} \rangle_{\mathbb{V}^*,\mathbb{V}} + \langle M_k^*\lambda + D_{du}Q_k(du, dp), \mu \rangle_{\mathbb{U}^*,\mathbb{U}} \\ + \langle g - G(du, dp, \lambda), \mu_{\text{dir}} \rangle_{\mathbb{P}^*,\mathbb{P}} + \langle g, d \rangle_{\mathbb{P}^*,\mathbb{P}}, \end{aligned}$$

where the new variable $g \in \mathbb{P}^*$ is introduced to play the role of the gradient of the reduced cost functional $\hat{Q}_k(dp)$. Similarly to the adjoint method, we use the identity:

$$\Phi_d(dp) = \mathcal{L}_2(S(dp), dp, S_{\text{adj}}(dp), \hat{G}(dp), \mu_{\text{adj}}, \mu, \mu_{\text{dir}}) \qquad \forall (\mu_{\text{adj}}, \mu, \mu_{\text{dir}}) \in \mathbb{V} \times \mathbb{U} \times \mathbb{P},$$

to compute the derivative of Φ_d. More precisely, we get

$$\begin{aligned} D\Phi_d(dp)\tilde{dp} = \left(D_{S(dp)}\mathcal{L}_2 \circ DS(dp)\right) + D_{dp}\mathcal{L}_2 + \left(D_{S_{\text{adj}}(dp)}\mathcal{L}_2 \circ DS_{\text{adj}}(dp)\right) \\ + \left(D_{\hat{G}(dp)}\mathcal{L}_2 \circ D\hat{G}(dp)\right), \end{aligned}$$

with a shortcut notation. Hence, choosing μ_{adj}, μ and μ_{dir} such that:

$$\begin{aligned} M_k^* \mu_{\text{adj}} &= -D^2_{du\,du} Q_k(du, dp)\mu + D^2_{du\,dp} Q_k(du, dp)\mu_{\text{dir}} \\ M_k \mu &= -F_k \mu_{\text{dir}} \\ \mu_{\text{dir}} &= -d \end{aligned}$$

ensures that

$$\begin{aligned} \left\langle D_{du}\mathcal{L}_2(du, dp, \lambda, g, \mu_{\text{adj}}, \mu, \mu_{\text{dir}}), \tilde{du} \right\rangle_{\mathbb{U}^*,\mathbb{U}} &= 0 \qquad \forall \tilde{du} \in \mathbb{U} \\ \left\langle D_{\lambda}\mathcal{L}_2(du, dp, \lambda, g, \mu_{\text{adj}}, \mu, \mu_{\text{dir}}), \tilde{\lambda} \right\rangle_{\mathbb{V}^*,\mathbb{V}} &= 0 \qquad \forall \tilde{\lambda} \in \mathbb{V} \\ \langle D_g\mathcal{L}_2(du, dp, \lambda, g, \mu_{\text{adj}}, \mu, \mu_{\text{dir}}), \tilde{g} \rangle_{\mathbb{P},\mathbb{P}^*} &= 0 \qquad \forall \tilde{g} \in \mathbb{P}^*. \end{aligned}$$

This allows us to write the differential of Φ_d as such:

$$D\Phi_d(dp) = -F_k^* \mu_{\text{adj}} + D^2_{dp\,du} Q_k(S(dp), dp)\mu - D^2_{dp\,dp} Q(S(dp), dp)\mu_{\text{dir}}.$$

Finally, according to the Riesz representation theorem, there exists a unique $H : \mathbb{P} \to \mathbb{P}$ such that

$$\left\langle D\Phi_d(dp), \tilde{dp} \right\rangle_{\mathbb{P}^*,\mathbb{P}} = \left\langle D^2\hat{Q}_k(dp)d, \tilde{dp} \right\rangle_{\mathbb{P}^*,\mathbb{P}} = \left(Hd, \tilde{dp} \right)_{\mathbb{P}} \qquad \forall d, \tilde{dp} \in \mathbb{P}.$$

Note that, since Q_k is quadratic, both $D^2_{du,du}Q_k$ and $D^2_{du,dp}Q_k$ are constant with respect to du and dp. Thus μ_{adj}, μ and μ_{dir} are not functions of du and dp. Hence, the reduced cost functional $\hat{Q}_k$ is quadratic.

Appendix B. evalutaing a TR-SQP step

As mentioned in Section 2.3, TR based algorithms require to evaluate both the predicted reduction $\text{pred}_k(dx)$ and the actual reduction $\text{ared}_k(dx)$ made by a step. In the case of unconstrained optimization problems, $\text{pred}_k(dx)$ is computed as the decrease of the quadratic approximation of the cost functional while $\text{ared}_k(dx)$ is computed as the decrease of the cost functional. However, when considering constrained optimization problems, simply reducing the cost function is not sufficient. We also need to reduce the model error. Moreover, we should keep in mind that, in some cases, it is necessary to increase the cost functional to reduce the model error. Conversely, a step that increases the model error but drastically decreases the cost functional can be considered to be a good step. To find a compromise, a *merit function* $\phi : \mathbb{X} \times \mathbb{R} \to \mathbb{R}$, which takes into account the value taken by the cost functional and the model error, is typically introduced. In the present work, we choose the following merit function inspired by Nocedal and Wright [7]:

$$\phi(x_k, \sigma_k) := \mathcal{J}(x_k) + \sigma_k \|\mathcal{M}(x_k) - f\|_{\mathbb{V}^*}, \tag{B.1}$$

where $\sigma_k \geq 0$ is a parameter that forces us to take the reduction of the model error into account. The actual reduction ared_k of the merit function at the step k can thus be computed as such:

$$\text{ared}_k(dx) := \phi(x_k, \sigma_k) - \phi(x_k + dx, \sigma_k). \tag{B.2}$$

The predicted reduction pred_k is typically derived by using a quadratic approximation of the cost functional $\mathcal{J}$ and a linear approximation of the model equation $\mathcal{M}$. This leads to the following definition:

$$\text{pred}_k(dx) := m_k(0) - m_k(dx) + \sigma_k \text{vpred}(k), \tag{B.3}$$

where $m_k : \mathbb{X} \to \mathbb{R}$ is the quadratic approximation of $\mathcal{J}$:

$$m_k(dx) = \mathcal{J}(x_k) + \langle D\mathcal{J}(x_k), dx\rangle_{\mathbb{X}^*,\mathbb{X}} + \frac{1}{2}\big\langle D^2\mathcal{J}(x_k)dx, dx\big\rangle_{\mathbb{X}^*,\mathbb{X}}$$

and where $\text{vpred}_k : \mathbb{X} \to \mathbb{R}$ is defined as such:

$$\text{vpred}_k(dx) := \|\mathcal{M}(x_k) - f\|_{\mathbb{V}^*} - \|\mathcal{M}(x_k) + D\mathcal{M}(x_k)dx - f\|_{\mathbb{V}^*}. \tag{B.4}$$

Note that solving (7) ensures that $\text{vpred}_k(dx_n) \geq 0$ and that solving (8) ensures that $\text{vpred}_k(dx_t) = 0$. It is thus possible to find a σ_k large enough so that $\text{pred}_k(dx)$ is positive, which is required. Moreover, we also impose that $\text{pred}_k(dx)$ is related to $\text{vpred}_k(dx)$ as follows:

$$\text{pred}_k(dx) \geq \tau\sigma_k \text{vpred}_k(dx), \tag{B.5}$$

where $\tau \in (0, 1)$. Typically, El-Alem takes $\tau = 0.5$, see [13, 15]. Moreover, the condition (B.5) can be ensured by setting:

$$\sigma_k = \frac{\langle D\mathcal{J}(x_k), dx\rangle_{\mathbb{X}^*,\mathbb{X}} + \frac{1}{2}\langle D^2\mathcal{J}(x_k)dx, dx\rangle_{\mathbb{X}^*,\mathbb{X}}}{(1-\tau)\text{vpred}_k(dx)}. \tag{B.6}$$

Standard TR-SQP algorithms starts with $\sigma_0 = 1$ and uses (B.6) to increase σ_k whenever (B.5) is not satisfied. However, it has been argued that keeping σ_k as small as possible increases the speed of convergence of the TR-SQP [12]. El-Alem proposes a *non-monotonic penalty parameter scheme*, which allows for diminishing σ_k when needed [15]. Note that, although the algorithm can find different local minima depending on how σ_k is updated, these minima, and thus the solution found by our algorithm, do not depend on σ_k, contrary to a penalty based method. However, if $\sigma_k = 0$ then every steps that do not decrease the cost functional will be rejected, which can prevent us from converging. Whereas, when $\sigma_k \to \infty$ every steps that do not decrease the model error is rejected. In such case, we loose the interest of working in an "extended search space". A good strategy is to start with $\sigma_0 = 1$ and to increase it during the optimization process, as discussed above. Hence, the choice of σ_k can influence both the convergence speed of the TR-SQP and its selected limit, but the minima found will not be a function of σ_k in itself.

Acknowledgments

This study was supported by the ANR AAPG program (project CLEARVIEW, ANR-17-CE23-0022). The research leading to these results also received funding from the European Union's Horizon 2020 research and innovation program under the ChEESE project, grant agreement No. 823844.

References

[1] Conn A, Gould N, Toint PL. Trust Region Methods, MPS/SIAM Ser. Optim, SIAM, Philadelphia. 2000;.

[2] Gould NI, Lucidi S, Roma M, Toint PL. Solving the trust-region subproblem using the Lanczos method. SIAM Journal on Optimization. 1999;9(2):504–525.

[3] Zhang LH, Shen C, Li RC. On the Generalized Lanczos Trust-Region Method. SIAM Journal on Optimization. 2017 jan;27(3):2110–2142. Available from: https://doi.org/10.1137%2F16m1095056.

[4] Steihaug T. The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. SIAM Journal on Numerical Analysis. 1983 jun;20(3):626–637. Available from: https://doi.org/10.1137%2F0720042.

[5] Lenders F, Kirches C, Potschka A. trlib: A vector-free implementation of the GLTR method for iterative solution of the trust region problem. Optimization Methods and Software. 2018;33(3):420–449.

[6] Moré JJ, Sorensen DC. Computing a trust region step. SIAM Journal on Scientific and Statistical Computing. 1983;4(3):553–572.

[7] Wright S, Nocedal J. Numerical optimization. Springer Science. 1999;35(67-68):7.

[8] Lalee M, Nocedal J, Plantenga T. On the implementation of an algorithm for large-scale equality constrained optimization. SIAM Journal on Optimization. 1998;8(3):682–706.

[9] Vicente LN. Trust-region interior-point algorithms for a class of nonlinear programming problems; 1996.

[10] Heinkenschloss M, Vicente LN. Analysis of inexact trust-region SQP algorithms. SIAM Journal on Optimization. 2002;12(2):283–302.

[11] Ziems JC, Ulbrich S. Adaptive multilevel inexact SQP methods for PDE-constrained optimization. SIAM Journal on Optimization. 2011;21(1):1–40.

[12] Gill PE, Murray W, Saunders MA, Wright MH. Some Theoretical Properties of an Augmented Lagrangian Merit Function. STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB; 1986.

[13] El-Alem M. A Global Convergence Theory for the Celis–Dennis–Tapia Trust-Region Algorithm for Constrained Optimization. SIAM Journal on Numerical Analysis. 1991 feb;28(1):266–290. Available from: https://doi.org/10.1137%2F0728015.

[14] Dennis JE, El-Alem M, Maciel MC. A Global Convergence Theory for General Trust-Region-Based Algorithms for Equality Constrained Optimization. SIAM Journal on Optimization. 1997 feb;7(1):177–207. Available from: https://doi.org/10.1137%2Fs1052623492238881.

[15] El-Alem M. A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization. SIAM Journal on Optimization. 1995;5(2):348–378.

[16] Lions JL. Contrôle optimal de systemes gouvernés par des équations aux dérivées partielles. Dunod; 1968.

[17] Chavent M. Analyse Fonctionelle et Identification de Coefficients Repartis dans les Equations aux Derivées Partielles. Thesis, Faculte des Science de Paris. 1971;.

[18] Le Dimet FX, Talagrand O. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. Tellus A: Dynamic Meteorology and Oceanography. 1986;38(2):97–110.

[19] Plessix RE. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. Geophysical Journal International. 2006;167(2):495–503.

[20] Le Dimet FX, Navon IM, Daescu DN. Second-order information in data assimilation. Monthly Weather Review. 2002;130(3):629–648.

[21] Yang P, Brossier R, Métivier L, Virieux J, Zhou W. A time-domain preconditioned truncated Newton approach to visco-acoustic multiparameter full waveform inversion. SIAM Journal on Scientific Computing. 2018;40(4):B1101–B1130.

[22] Métivier L, Brossier R, Virieux J, Operto S. Full waveform inversion and the truncated Newton method. SIAM Journal on Scientific Computing. 2013;35(2):B401–B437.

[23] Van Leeuwen T, Herrmann FJ. Mitigating local minima in full-waveform inversion by expanding the search space. Geophysical Journal International. 2013;195(1):661–667.

[24] van Leeuwen T, Herrmann FJ. A penalty method for PDE-constrained optimization in inverse problems. Inverse Problems. 2015 dec;32(1):015007. Available from: https://doi.org/10.1088%2F0266-5611%2F32%2F1%2F015007.

[25] Aghamiry H, Gholami A, Operto S. Improving full-waveform inversion based on wavefield reconstruction via Bregman iterations. In: 80th EAGE Conference and Exhibition 2018; 2018. .

[26] Aghamiry HS, Gholami A, Operto S. Improving full-waveform inversion by wavefield reconstruction with the alternating direction method of multipliers. Geophysics. 2019;84(1):R139–R162.

[27] Bermúdez A, Hervella-Nieto L, Prieto A, Rodrı R, et al. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. Journal of Computational Physics. 2007;223(2):469–488.

[28] Bernard S, Monteiller V, Komatitsch D, Lasaygues P. Ultrasonic computed tomography based on full-waveform inversion for bone quantitative imaging. Physics in Medicine & Biology. 2017 aug;62(17):7011–7035. Available from: https://doi.org/10.1088%2F1361-6560%2Faa7e5a.

[29] Bachmann E, Tromp J. Source encoding for viscoacoustic ultrasound computed tomography. The Journal of the Acoustical Society of America. 2020;147(5):3221–3235. Available from: https://doi.org/10.1121/10.0001191.

[30] Faucher F, Scherzer O, Barucq H. Eigenvector models for solving the seismic inverse problem for the Helmholtz equation. Geophysical Journal International. 2020 jan;221(1):394–414. Available from: https://doi.org/10.1093%2Fgji%2Fggaa009.

[31] de Buhan M, Kray M. A new approach to solve the inverse scattering problem for waves: combining the TRAC and the adaptive inversion methods. Inverse Problems. 2013 jul;29(8):085009. Available from: https://doi.org/10.1088%2F0266-5611%2F29%2F8%2F085009.

[32] Kaltenbacher B. Regularization Based on All-At-Once Formulations for Inverse Problems. SIAM Journal on Numerical Analysis. 2016 jan;54(4):2594–2618. Available from: https://doi.org/10.1137%2F16m1060984.

[33] Feng D, Pulliam TH. An all-at-once reduced Hessian SQP scheme for aerodynamic design optimization. 1995;.

[34] Kim HK, Gu X, Hielscher AH. An all-at-once reduced Hessian SQP algorithm for frequency domain optical tomography. In: Tromberg BJ, Yodh AG, Tamura M, Sevick-Muraca EM, Alfano RR, editors. Optical Tomography and Spectroscopy of Tissue VIII. SPIE; 2009. Available from: https://doi.org/10.1117%2F12.809385.

[35] Heinkenschloss M, Ridzal D. An Inexact Trust-Region SQP Method with Applications to PDE-Constrained Optimization. In: Numerical Mathematics and Advanced Applications. Springer Berlin Heidelberg;. p. 613–620. Available from: https://doi.org/10.1007%2F978-3-540-69777-0_73.

[36] SHENOY A, HEINKENSCHLOSS M, CLIFF EM. Airfoil Design by an All-at-once Method. International Journal of Computational Fluid Dynamics. 1998 nov;11(1-2):3–25. Available from: https://doi.org/10.1080%2F10618569808940863.

[37] Epanomeritakis I, Akçelik V, Ghattas O, Bielak J. A Newton-CG method for large-scale three-

dimensional elastic full-waveform seismic inversion. Inverse Problems. 2008 may;24(3):034015. Available from: https://doi.org/10.1088%2F0266-5611%2F24%2F3%2F034015.