Quantitative localization of small obstacles with single-layer potential fast solvers

Ha Pham, Hélène Barucq, Florian Faucher

To cite this version:

Ha Pham, Hélène Barucq, Florian Faucher. Quantitative localization of small obstacles with single-layer potential fast solvers. [Research Report] Inria Bordeaux Sud-Ouest; Magique 3D; Universite de Pau et des Pays de l'Adour. 2017. hal-01673475v1

HAL Id: hal-01673475
https://inria.hal.science/hal-01673475v1

Submitted on 29 Dec 2017 (v1), last revised 9 Jan 2018 (v2)

Quantitative localization of small obstacles with single-layer potential fast solvers

Hélène Barucq, Florian Faucher, Ha Pham

RESEARCH REPORT

N° 9137

December 2017

Project-Team Magique 3D

ISSN 0249-6399 ISRN INRIA/RR--9137--FR+ENG

Quantitative localization of small obstacles with single-layer potential fast solvers

Hélène Barucq*, Florian Faucher*, Ha Pham*

Project-Team Magique 3D

Research Report n° 9137 — December 2017 — 74 pages

Abstract: In this work, we numerically study the inverse problem of locating small circular obstacles in a homogeneous medium using noisy backscattered data collected at several frequencies. The main novelty of our work is the implementation of a single-layer potential based fast solver (called FSSL) in a Full-Waveform inversion procedure, to give high quality reconstruction with low-time cost. The efficiency of FSSL was studied in our previous works [7, 6]. We show reconstruction results with up to 12 obstacles in structured or random configurations with several initial guesses, all allowed to be far and different in nature from the target. This last assumption is not expected in results using nonlinear optimization schemes in general. For results with 6 obstacles, we also investigate several optimization methods, comparing between nonlinear gradient descent and quasi-Newton, as well as their convergence with different line search algorithms.

Key-words: Full-waveform inversion, locating small obstacles, inverse multiple-scattering problem, single-layer potential, quantitative inversion, multiple-frequency, frequency-hopping, backscattered data, adjoint-state method, Helmholtz equation, reconstruction, nonlinear gradient-based optimization.

---

* Inria Bordeaux Sud-Ouest, Project-Team Magique 3D, Université de Pau et des pays de l'Adour, France

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Localisation de petits obstacles avec un solveur rapide utilisant une méthode de potentiel de simple couche

Résumé : Dans ce travail, nous étudions le problème inverse pour la localisation d'obstacles circulaires dans un milieu homogène en utilisant des données multi-fréquences de type 'backscattering'. La principale nouveauté de ce travail est l'implémentation d'un solveur rapide utilisant les potentiels simple couche (dénommé FSSL) au sein d'une procédure d'inversion des formes d'onde (FWI), afin d'obtenir une reconstruction de grande qualité, à un faible coût numérique. L'efficacité de la méthode FSSL pour le problème direct a été étudiée dans nos précédent travaux, [7, 6]. Ici, nous montrons des reconstructions jusqu'à 12 obstacles dans des configurations structurées ou aléatoires, avec plusieurs modèles initiaux éloignés et de nature différente par rapport à la cible. Dans le cas de reconstruction de 6 obstacles, nous présentons aussi une étude des différentes méthodes d'optimisation implémentées dans notre code, en comparant le gradient non-linéaire conjugué et quasi-Newton, ainsi que la convergence de différentes méthodes de line search.

Mots-clés : Inversion des formes d'onde, Full-waveform inversion, localisation de petits obstacles, problème inverse en multiple-scattering, potentiel simple couche, inversion quantitative, propagation harmonique, multi-fréquences, méthode de l'état adjoint, équation de Helmholtz, reconstruction, optimisation non-linéaire.

1 Introduction

In this work, we consider the inverse problem of locating quantitatively small circular obstacles in a homogeneous medium using multi-frequency backscattered data. This inverse problem has applications in materials imaging e.g. non-destructive testing using acoustic waves to detect defects, buried objects location, geophysical exploration and medical imaging, c.f. [2, 10]. The general framework of an inverse problem starts with the forward map Φ which assigns to a model $\boldsymbol{p}$ in a model space $\boldsymbol{\mathcal{P}}$ observed dataset (state) $\mathbf{d}$ in a state space $\boldsymbol{\mathcal{D}}$. An inverse problem aims to reconstruct the unknown model which gives rise to an observed data $\mathbf{d}_{\text{obs}}$, i.e. one tries to solve for $\Phi^{-1}(\mathbf{d}_{\text{obs}})$.

The unknowns in our inverse problems are the positions of the obstacles and the observed data are noisy synthetic back-scattered data. We assume that the number, size and scattering type (soft or hard) of the obstacles are known, and that the obstacles are non-overlapping and located strictly inside a rectangular domain of interest in which experiments will be carried out to collect the back-scattered data (along the boundaries of the domain). The inverse problem is written as a nonlinear model-based minimization problem and solved by gradient-based optimization, a procedure in which an initial guess is corrected in hope of converging towards the true model. In seismic inversion, this approach is called Full Waveform Inversion (FWI), see e.g. [46, 42, 19]. FWI using only gradient-based optimization with simple linesearch has been effective in large-scale optimization, in reconstructing velocity of complex media[1]. With increasing computing powers, FWI has become a popular tool in seismic inversion, however the method enjoys much less popularity in smaller-scale problems. For this reason, we would like to investigate the performance of FWI in the context of inverse scattering. The main novelty of our work is the implementation of a fast solver in FWI procedure to give high quality reconstruction with low-time cost, this is further discussed below.

For the localization problem, there are two approaches in describing the direct problem mathematically. In our work, the direct phenomenon is modeled as a multiple-obstacle scattering and mathematically described as an exterior boundary value problem of the Helmholtz equation, with boundary condition imposed on the obstacles and radiation condition imposed at infinity, see discussion in Section 2. Also in our approach, the position of the obstacles are retrieved directly, and the typical data set are either far-field or near-field data. Another viewpoint is to consider the obstacles as part of the homogeneities of the medium, e.g. in Electrical Impedance Tomography (EIT) in biomedical imaging, the obstacles are represented by regions having different conductivity and the dataset in this setting being the Dirichlet-to-Neumann map (D-t-N)[2], see e.g. [2], or in inhomogeneous medium scattering, as regions having different value of refractive index[3].

Reconstruction methods fall into two categories: quantitative, which includes FWI, and qualitative (also called imaging-based direct methods). Quantitative inversion has the advantage of being ‘conceptually simple’ and giving better precision reconstruction [34]; however, these procedures require solving numerous direct and adjoint problems, and can thus be computationally intensive. Most works in inverse scattering focus on shape reconstruction (of extended obstacles)

---

[1] Numerical reconstruction of synthetic examples modeling Earth’s subsurfaces can be found e.g. [17]

[2] The direct problem is now modeled as $\nabla \cdot (\gamma(x)\nabla u) = 0$ with the conductivity $\gamma(x)$ being constant outside of the obstacles and taking extreme values inside the obstacles. Under small inhomogeneities assumption see e.g. [21] for a description of γ. This equation is coupled with a Dirichlet boundary condition on the boundary of the domain of interest, which in the context of EIT is a current applied to the boundary. The problem is in a finite domain rather than an infinite one as in multiple-obstacle scattering. In the context of EIT, the D-t-N gives the potential, measured on the boundary of the domain of interest, induced by currents also applied on the boundary.

[3] This perspective is described by $-\Delta - \kappa^2 n(x) = 0$ and radiation condition is imposed at infinity. The information of the obstacles are now contained in the refractive index $n(x)$. The presence of an obstacle is signaled by n taking on different value from the background medium, see e.g. [13, 24].

or on localization of scatterers as inhomogeneities (the EIT pointview); due to the computational burden of quantitative inversion, they use imaging-based methods (such as direct sampling, factorization, MUSIC, probe methods etc). For examples of qualitative reconstruction of shape, see the survey paper [47], book [10], or the introduction of [34, 38, 5] and the references therein. For localization problems using qualitative methods, see e.g. [18, 1, 28, 11, 24, 16].

The computational cost of quantitative inversion can be drastically reduced by using fast solvers (for the resolution of forward problems). In previous works [7, 6], we have investigated the efficiency of a solver based on single-layer potential (called FSSL) in simulating the diffraction by a large number of small soft or hard-scattering obstacles, in comparison to methods based on volume discretization e.g. Finite Element methods. This solver reduces drastically the resolution time-cost of a forward problem and hence the overall time-cost of a FWI procedure. The second advantage of FSSL method is the high-precision simulation of the forward problem, and as a result, the quality of the final reconstruction[4]. In dealing with small scatterers or homogeneities, a reductive model is usually employed by asymptotic expansion (see e.g. [2]), or the interaction between the obstacles is approximated by single-scattering (also called Born approximation), or the scatterers are considered point-like (e.g. in Foldy-Lax model [28]). This is not the case for our solver which takes into consideration all levels of interaction between the scatterers, and can treat both small and extended scatterers[5]. Comparable to our approach for the direct problem is the generalized Foldy-Lax model used in [4] and the references therein, however a different linear equation was used[6] and direct-imaging is used for inversion in [4]. Another advantage of the single-layer potential solver is that it does not require domain truncation, which is typical of integral equation methods, which makes application in FWI simplier and gives higher precision modeling, compared to using FWI with volume-based discretization.

The literature on quantitative inverse scattering focuses mostly on shape reconstruction, see e.g. [36, 33, 35, 31] or inhomogeneities of the medium [8] and the references therein. While FWI is popular in seismic inversion, see e.g. [46], few literature can be found on inverse scattering using true FWI approach, with the exception of the recent work [20]. However, there are several differences between [20] and our paper: 1. the obstacles are characterized as part of the medium in terms of speed, density of medium (versus to our unknown parameters being exactly the position of the obstacles), 2. the linear problem is simplified with Born approximation and, 3. the direct problem is discretized and numerically solved by volumed-based methods (finite difference or finite element), 4. FWI is carried out in the time-domain for the wave equation describing a medium with absorption, 5. numerical experiments are shown for up to 5 obstacles for structured configurations.

Other novelties in our experiments are the usage of back-scattered data and the fact that our initial guesses are allowed to be far and different in nature from the target, a property not expected of methods based on nonlinear optimization schemes in general. Regarding dataset, most references work with the far-field pattern or Dirichlet-to-Neumann map as scattered data rather than backscattered data (with the exception of e.g. [24] which however uses Music-type inversion). Back-scattered data are produced by a single sensory device (which acts as an array of

---

[4] The precision in forward problem resolution, an important point in FWI, helps reduce the number of artificial local minima of the cost function.

[5] For circular obstacles, to guarantee the invertibility of the forward problem, the testing wavenumber κ and the radius of the obstacles r have to satisfy: $\kappa\mathsf{r}$ are not in the set of zeros of Bessel functions. A 'cheap' requirement however enough for our tests is that $\kappa\mathsf{r} \leq 2$, for more details see [7, Remark 2]

[6] In [4], single-layer potential Ansatz is also used to distinguish the scattered field by extended obstacles from that by the point-like scatterers. However, a different choice of boundary operators (using a combination of interior trace operators) result in the so-called Combined Field Field Integral Equation [4, Eqn 2.13]. For more details on the derivation of this equation see e.g. [3, Eqn 64]. These equations are more complicated (containing more terms) than the ones that we use, which are obtained simply by applying the exterior traces (for both the Dirichlet and Neumann problems) to the Ansatz [7].

transducer and as well as receivers)[7]. Regarding initial guesses, most iterative-based inversions are said to require sufficient closeness of the initial guesses to the true target [32]. In our experiments, despite the restriction on the angle of data collection, with the usage of multiple-frequency[8], our initial guesses are allowed to be far and different in nature from the target. In addition, we show results with up to 12 obstacles in structured or random configurations, these are on the higher end in terms of the number of obstacles that can be detected among results using either quantitative or qualitative methods. Another contribution of our work is the investigation of several optimization methods, comparing between nonlinear gradient descent (a popular choice in seismic inversion) and quasi-Newton, as well as their convergence with different linesearch algorithms. As pointed out in [34], although the implementation of quantitative inversion is 'simpler' in general, the theoretical side of the convergence and uniqueness in inverse multiple-scattering is not completely settled, see Remark 1. These experiments are contributions to future theoretical investigation of the convergence and uniqueness of quantitative methods in inverse multiple-scattering.

The remaining of the paper is organized as follows. Section 2 introduces the fast solver which is used to discretize the direct problem, and the discrete inverse problem as an optimization problem of the cost function. The key ingredient in gradient-based optimization is the Fréchet derivative of the cost function, which in our paper is computed using the Adjoint-State method, from two perspectives. This is the subject of Section 3 with further details in Appendix B. In Section 4, the frequency-hopping procedure with the investigated search directions and line search algorithms are described. Section 5 serves as an overall discussion and highlights the common theme of the numerical experiments follows after. The numerical experiments are given in Section 6 for 6 obstacles, 7 for 12 obstacles in structured configuration, and Section 8 for 12 obstacles in random configuration. Section 8 aims to show that the inversion procedure also works for arbitrary configurations.

Remark 1. *The shape and location for one obstacle can be uniquely determined from the far-field pattern [23, 14]. Lipschitz stability study of the forward map Φ, and the convergence for gradient-based and Newton-like iterations are carried out for shape reconstruction with the scattered data being far-field pattern, see e.g. [43, 29]. When the obstacles are considered as part of the inhomogeneities (or conductivities) of the medium, and the dataset are the Dirichlet-to-Neumann operators (also called boundary measurements), uniqueness and stability estimates are given by [21]. There is also a uniqueness result with respect to the location of point-like homogeneities by [22]. Uniqueness with the scattered data being far-field pattern with respect to the location of homogeneities of finite refractive index and under smallness in size assumption is given in [24].*

2 Discrete direct and inverse problem

We first describe how the direct problem is discretized by the single-layer potential fast solver. For convenience, this is also called FSSL.

Notations Consider a configuration of N_{Obs} non-overlapping circular obstacles of the same radius. We assume the circular obstacle I is of radius r_I and centered at $\mathbf{x}_I \in \mathbb{R}^2$. located at

$$\mathbf{x}_I = (\mathbf{x}_{I,1}, \mathbf{x}_{I,2}) \ , \ 1 \leq I \leq N_{\text{Obs}} \, .$$

---

[7] This type of partial data (versus full data which are collected all around the domain of interest or at wider angles) which is not compatible with traditional MUSIC-type methods [24], also this represents more challenges for the reconstruction, see also discussion in Section 5.

[8] Multi-frequency data is a natural assumption in physical experiments [24]

The configuration is determined by vector $\boldsymbol{p}$, called the parameter vector of size $N_{\text{par}} = 2 \times N_{\text{Obs}}$.

$$\boldsymbol{p} = \left(\mathbf{x}_{1,1}, \mathbf{x}_{1,2} \; \ldots, \mathbf{x}_{N_{\text{Obs}},1}\,,\, \mathbf{x}_{N_{\text{Obs}},2}\right).$$

Denote the polar coordinates relative to $\mathbf{x}_I$ by $(r_I(\cdot), \theta_I(\cdot))$,

$$x \;=\; \mathbf{x}_I \,+\, r_I(x)\,\big(\cos\theta_I(x)\,,\, \sin\theta_I(x)\big)\,. \tag{1}$$

Denote by d_{IJ} the distance between the centers of obstacle I and J, and by θ_{IJ} and θ_{JI} their relative polar coordinates,

$$\mathbf{x}_I = \mathbf{x}_J + \mathrm{d}_{IJ}\,(\cos\theta_{JI}\,,\, \sin\theta_{JI})\,; \quad \mathbf{x}_J = \mathbf{x}_I + \mathrm{d}_{IJ}\,(\cos\theta_{IJ}\,,\, \sin\theta_{IJ})\,.$$

The non-overlapping assumption is given by $\mathrm{d}_{IJ} > \mathrm{r}_I + \mathrm{r}_J$.

The continuous forward problem In nondestructive experiments to detect the location of obstacles, incident waves are sent into the domain of interest to be diffracted by the obstacles, and corresponding scattered waves u are recorded at receivers, see Figure 1. In particular, for a time-harmonic acoustic wave $u_{\text{inc}}(x)\, e^{\mathrm{i}2\pi \mathrm{f}\, t}$ (of frequency f), propagating in a homogeneous medium with wave speed c, the scattered wave u is required to solve the Helmholtz equation in the domain outside the obstacles defined by Ω_{ext},

$$(-\Delta - \kappa^2)\, u \;=\; 0\,,$$

and to satisfy the κ-outgoing condition at infinity to ensure that it does not re-enter the domain of interest

$$\lim_{r\to\infty} \sqrt{r}\,(\partial_r u - \mathrm{i}\,\kappa\, u) = 0 \quad , \quad r = |x|\,.$$

Here, wavenumber κ is determined by the dispersion relation $\kappa = \frac{2\pi \mathrm{f}}{c}$ and the corresponding wavelength λ is given by $\lambda = \frac{2\pi}{\kappa}$.

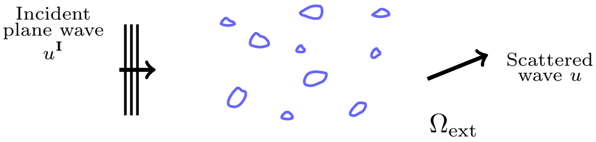


Figure 1: Scattering of a plane wave by various and non-overlapping obstacles. Ω_{ext} is the domain outside the obstacles.

We will be working mainly with incident wave in the form of time-harmonic planewave, characterized by the angle of incidence (also called angle of illumination) α_{inc}.

$$u_{\text{pw}}(\mathbf{x}) \quad = \quad e^{\mathrm{i}\,\kappa \mathbf{x}\cdot(\cos\alpha_{\text{inc}}, \sin\alpha_{\text{inc}})}\,. \tag{2}$$

The number of the angle of incidence is denoted by N_{Acq}.

We assume that the obstacles are either soft- or hard- scattering, upon which Dirichlet or Neumann boundary conditions are imposed on their boundaries,

$$\begin{array}{ll} \gamma_0^+(u + u_{\text{inc}}) = 0 & \text{for Dirichlet condition ,} \\ \gamma_1^+(u + u_{\text{inc}}) = 0 & \text{for Neumann condition} \end{array} \quad \text{along } \Gamma_{\text{Obs}}\,.$$

See [7] for the definition of the exterior trace operators γ_0^+ and γ_1^+.

Discretization of the forward problem by FSSL discretization We list the main idea of FSSL, for more details see [7, 6]. In this method, the scattered response is written as a superposition of waves scattered by each obstacle which are in the form of acoustic Single-Layer Potentials. In this way, the Helmholtz equation and the radiation condition are satisfied automatically. When there are N_{Obs} non-overlapping obstacles Ω_I with boundary denoted by Γ_I, $1 \leq I \leq N_{\text{Obs}}$, we write

$$u(x) \;=\; \sum_{I=1}^{N} (\mathcal{S}_I\, \tilde{v}_I)(x) \;:=\; \sum_{I=1}^{N} \int_{\Gamma_I} G_\kappa(x,y)\, \tilde{v}_I(y)\, dy \quad , \quad \tilde{v}_I \in \mathcal{C}(\Gamma_I). \tag{3}$$

Here G_κ is the outgoing fundamental solution at wavenumber κ, and is given in terms of the Hankel function of the first kind $\mathrm{H}_0^{(1)}$,

$$G_\kappa(x,y) \;:=\; \tfrac{\mathrm{i}}{4}\, \mathrm{H}_0^{(1)}(\kappa\, |x-y|) \quad , \quad x \neq y\,. \tag{4}$$

The unknowns of the multi-scattering problem are now the single-layer densities $\tilde{v}_J$. The discretized scattered waves are obtained by a truncation of the Fourier Series of the single-layer densities. For discretization method of order $\mathbf{m}$, the approximate diffractive field is written as a linear combination of single-layer potentials with approximate single-layer densities $\{v_{h,J}\}$, using $2\mathbf{m}+1$ Fourier nodes between $-\mathbf{m}$ and $\mathbf{m}$.

$$u_h \;=\; \sum_{J=1}^{N_{\text{Obs}}} \mathcal{S}_J\; v_{h,J}\ , \text{with} \quad v_{h,J}(\theta_J(\mathbf{x})\,,\, r_J(\mathbf{x})) \;=\; \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{J,l}\; e^{\mathrm{i}\, l\, \theta_J(\mathbf{x})}\,. \tag{5}$$

The coefficients $V = (V_{J,l})_{1 \leq J \leq N_{\text{Obs}}, -\mathbf{m} \leq l \leq \mathbf{m}}$ are the coefficients of the generalized Fourier series for $v_{h,J}$. We can rewrite (5) in a more compact form using matrix-vector multiplication

$$u_h(\mathbf{x}) \;:=\; T(\boldsymbol{p})^t \underset{\substack{\uparrow \\ \text{matrix} \\ \text{multiplication}}}{\qquad} V. \tag{6}$$

Here $T(\boldsymbol{p})$ is a function-valued column vector composed of N_{Obs} blocks and each block is of size $2\mathbf{m}+1$

$$T \quad : \quad \boldsymbol{\mathcal{P}} \quad \longrightarrow \quad \mathcal{M}_{N\times 1}(\mathcal{C}(\mathbb{R}^2;\mathbb{C})) \quad ; \quad \big(T(\boldsymbol{p})\big)_{Jl} = \mathcal{S}_J(\boldsymbol{p})(\mathbf{w}_{Jl})\,. \tag{7}$$

Here, $\mathcal{M}_{n\times m}(\mathbb{F})$ denotes the space of matrices of size $n \times m$ with components taking value in $\mathbb{F}$.

When the obstacles are circular, the scattered wave can be written as multipole expansion defined in terms of Hankel functions of the first kind $\mathrm{H}_k^{(1)}$,

$$u_h(\mathbf{x}) \;=\; \frac{\mathrm{i}\,\pi}{2} \sum_{J=1}^{N_{\text{Obs}}} \mathsf{r}_J \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{J,l}\; \mathrm{H}_k^{(1)}(\kappa\, r_J(\mathbf{x}))\; e^{i\, l\, \theta_J(\mathbf{x})}\,. \tag{8}$$

In its compact form (6), the function-valued column vector $T(\boldsymbol{p})$ defined in (7) is given by,

$$T(\boldsymbol{p})_{J,l} \;=\; \frac{\mathrm{i}\,\pi}{2} \mathsf{r}_J\; \mathrm{H}_k^{(1)}(\kappa\, r_J(\mathbf{x}))\; e^{\mathrm{i}\, l\, \theta_J(\mathbf{x})}. \tag{9}$$

The unknowns V solve the multiple-scattering linear system

$$\mathbf{A}(\boldsymbol{p})\, V \;=\; \quad F(\boldsymbol{p}, \alpha_{\text{inc}}). \tag{10}$$

with mapping properties,

$$\begin{aligned} V\ ,\ F \quad &: \quad \boldsymbol{\mathcal{P}} = \mathbb{R}^{N_{\text{par}}} \quad \longrightarrow \quad \mathcal{M}_{N\times 1}(\mathbb{C})\,; \\ A \quad &: \quad \boldsymbol{\mathcal{P}} = \mathbb{R}^{N_{\text{par}}} \quad \longrightarrow \quad \mathcal{M}_{N\times N}(\mathbb{C}) \quad . \end{aligned}$$

That V is a solution of (10) is equivalent to

$$V \;=\; \mathsf{S}(\boldsymbol{p}, \alpha_{\text{inc}})\,.$$

where solution operator S is defined as

$$\mathsf{S}\ :\ \boldsymbol{\mathcal{P}} \quad \longrightarrow \quad \mathbb{C}^N \quad , \quad \boldsymbol{p} \quad \mapsto \quad \mathbf{A}(\boldsymbol{p})^{-1}\, F(\boldsymbol{p}, \alpha_{\text{inc}}). \tag{11}$$

The multi-scattering matrix $\mathbf{A}$ composes of $N_{\text{Obs}} \times N_{\text{Obs}}$ block matrices, each of which is a matrix of size $(2\mathbf{m}+1) \times (2\mathbf{m}+1)$.

$$\mathbf{A}_\alpha \;=\; \begin{pmatrix} \mathbf{A}_{\alpha\,;\,1} & \mathbf{A}_{\alpha\,;\,12} & \dots & \mathbf{A}_{\alpha\,;\,1\,(N-1)} & \mathbf{A}_{\alpha\,;\,1\,N} \\ \mathbf{A}_{\alpha\,;\,21} & \mathbf{A}_{\alpha\,;\,2} & \dots & \mathbf{A}_{\alpha\,;\,2\,(N-1))} & \mathbf{A}_{\alpha\,;\,2\,N} \\ \vdots & \dots & \dots & \ddots & \vdots \\ \mathbf{A}_{\alpha\,;\,(N-1)\,1} & \mathbf{A}_{\alpha\,;\,(N-1)\,2} & \dots & \mathbf{A}_{\alpha\,;\,N-1} & \mathbf{A}_{\alpha\,;\,(N-1)N} \\ \mathbf{A}_{\alpha\,;\,N\,1} & \mathbf{A}_{\alpha\,;\,N\,2} & \dots & \mathbf{A}_{\alpha\,;\,N\,(N-1)} & \mathbf{A}_{\alpha\,;\,N} \end{pmatrix}.$$

The diagonal operator $\mathbf{A}_{\alpha\,;\,I}$, describes self-reflection by obstacle I. The off-diagonal $\mathbf{A}_{\alpha\,;\,IJ}$ with $I \neq J$ describes the diffraction by obstacle I of the wave emitted by J. The size of the linear system is given by

$$N = (2\mathbf{m}+1)N_{\text{Obs}} \quad .$$

The invertibility of the system is discussed in Footnote 5.

For circular obstacles, the multiple-scattering linear system can be described explicitly in the form of multipole expansion, using the Hankel functions of the first kind $\mathrm{H}_k^{(1)}$ and the Bessel function J_k and their derivatives. The diagonal blocks $\mathbf{A}_{\alpha,I}$ are diagonal matrices, with diagonal components given by

$$(\mathbf{A}_I)_{ll} = i\,\pi\,\mathsf{r}_I\,\mathrm{J}_l(\kappa\,\mathsf{r}_I) \ \times \begin{cases} \mathrm{H}_l^{(1)}(\kappa\,\mathsf{r}_I) & , \quad \text{for soft-scattering (Dirichlet BC)} \\ \kappa\,\mathrm{H}_l^{(1)\prime}(\kappa\,\mathsf{r}_I) & , \quad \text{for hard-scattering (Neumann BC)} \end{cases}. \tag{12}$$

For $I \neq J$, the components of the off-diagonal block $\mathbf{A}_{\alpha,IJ}$ are given by,

$$\begin{aligned} (\mathbf{A}_{\alpha,IJ})_{lm} = i\,\pi\,\mathsf{r}_J \mathrm{J}_m(\kappa\,\mathsf{r}_J)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ})\ e^{i(m-l)\boldsymbol{\theta}_{JI}} \\ \times \begin{cases} \mathrm{J}_l(\kappa\,\mathsf{r}_I) & , \quad \text{for soft-scattering} \\ \kappa\,\mathrm{J}_l'(\kappa\,\mathsf{r}_I) & , \quad \text{for hard-scattering} \end{cases}. \end{aligned} \tag{13}$$

The components of the RHS corresponding to planewave u_{pw} (2) are given by

$$F_{I,l} \;= -\,2\,u_{\text{pw}}(\mathbf{x}_I)\,\mathrm{i}^l\ e^{-\mathrm{i}\,l\,\alpha_{\text{inc}}} \times \begin{cases} \mathrm{J}_l(\kappa\,\mathsf{r}_I) & , \quad \text{for soft-scattering} \\ \kappa\,\mathrm{J}_l'(\kappa\,\mathsf{r}_I) & , \quad \text{for hard-scattering} \end{cases}. \tag{14}$$

In short, the discretization of the forward problem in the case of circular obstacles can be summarized as

$$\begin{aligned} u_h(\mathbf{x}) \;&=\; T(\boldsymbol{p})^t\, V \quad ; \\ \mathbf{A}(\boldsymbol{p})\, V \;&=\; F(\boldsymbol{p}, \alpha_{\text{inc}}) \quad \Leftrightarrow \quad V \;=\; \mathsf{S}(\boldsymbol{p}, \alpha_{\text{inc}})\,, \end{aligned} \tag{15}$$

with (column) vectors $\in \mathbb{C}^N$ $T(\boldsymbol{p})$ defined in (7), $V(\boldsymbol{p})$ and $F(\boldsymbol{p})$ defined (14), and matrix $\mathbf{A}(\boldsymbol{p})$ defined (12) .

The discrete forward map Φ Consider a set of receivers located at $\mathsf{R}_1, \ldots, \mathsf{R}_{N_{\text{rec}}} \in \mathbb{R}^2$. Denote by $\mathcal{R}$ the linear evaluation (restriction) of a function to this set. $\mathcal{R}_{\text{rec}}$ acts on column vectors, and $\tilde{\mathcal{R}}$ acts on row vectors,

$$\begin{aligned} \mathcal{R} \quad : \quad \mathcal{M}_{n\times 1}\big(\mathcal{C}(\Omega_{\text{rec}};\mathbb{C})\big) \quad &\longrightarrow \quad \mathcal{M}_{n\times N_{\text{rec}}}(\mathbb{C}) \\ v = (v_i) \quad &\longmapsto \quad (\mathcal{R}\, v)_{i,k} \;=\; v_i(\,\mathsf{R}_k) \quad ; \\ \tilde{\mathcal{R}} \quad : \quad \mathcal{M}_{1\times n}\big(\mathcal{C}(\Omega_{\text{rec}};\mathbb{C})\big) \quad &\longrightarrow \quad \mathcal{M}_{N_{\text{rec}}\times n}(\mathbb{C}) \quad . \end{aligned} \tag{16}$$

Here $\mathcal{C}(\Omega_{\text{rec}};\mathbb{C})$ is the space of complex-valued functions of real variables $\mathbb{R}^2$ continuous in a neighborhood $\Omega_{\text{rec}} \subset \mathbb{R}^2$ of the receivers.

For a given configuration $\boldsymbol{p}$, we define the **forward map** Φ which gives the simulated data, which is the field scattered by the configuration, recorded at the receivers positioned at $\mathsf{R}_1, \ldots, \mathsf{R}_{N_{\text{rec}}} \in \mathbb{R}^2$,

$$\begin{aligned} \Phi \quad : \quad \boldsymbol{\mathcal{P}} = \mathbb{R}^{N_{\mathbf{par}}} \qquad &\longrightarrow \qquad \boldsymbol{\mathcal{D}} = \mathbb{C}^{N_{\text{rec}}} \\ \boldsymbol{p} \quad &\mapsto \quad \Phi(\boldsymbol{p}) \;:=\; \mathcal{R}\, T(\boldsymbol{p})^t\, \mathrm{S}(\boldsymbol{p}) = \left(\sum_{J=1}^{N_{\text{Obs}}} \sum_{l\in\mathbb{Z}} v_{J,l} \, \Big(\mathcal{S}_J \mathbf{w}_{J,l}\Big)(\mathsf{R}_k) \right)_{1\le k\le N_{\text{rec}}} \end{aligned} \tag{17}$$

It is convenient to use

$$\mathcal{R}\, \Big(T(\boldsymbol{p})^t\; V\Big) \;=\; \Big(\tilde{\mathcal{R}}\, T(\boldsymbol{p})^t\Big)\; V$$

which applies the restriction operator to the basis functions first and the object $\tilde{\mathcal{R}}\, T(\boldsymbol{p})^t$ is a matrix of size $N_{\text{rec}} \times N$ with row indexed by k with $1 \le k \le N_{\text{rec}}$ and the columns indexed by (J, l) with $1 \le J \le N_{\text{Obs}}$, $-\mathbf{m} \le l \le \mathbf{m}$. We denote this matrix by $\boldsymbol{\mathfrak{R}}$.

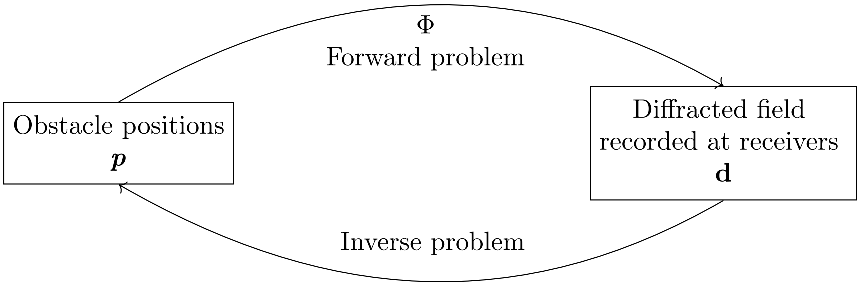
Its component at row k and column (J, l) is given by

$$\Big(\boldsymbol{\mathfrak{R}}(\boldsymbol{p})\Big)_{k,(J,l)} \;=\; \Big(\tilde{\mathcal{R}}\, T(\boldsymbol{p})^t\Big)_{k,(J,l)} \;=\; \big(\mathcal{S}_J\, \mathbf{w}_{J,l}\big)\,(\mathsf{R}_k). \tag{18}$$

We write

$$\Phi(\boldsymbol{p}) \;=\; \boldsymbol{\mathfrak{R}}(\boldsymbol{p})\; \mathsf{S}(\boldsymbol{p})\,, \tag{19}$$

Discrete inverse problem Given the observed dataset vector $\mathbf{d}_{\text{obs}} \in \boldsymbol{\mathcal{D}}$, the inverse problem aims to find the configuration $\boldsymbol{p}$ that gave rise to these data.



In FWI approach, the discrete inverse problem is written as an unconstrained optimization problem of the reduced cost function $\widehat{J}$

$$\boxed{\min_{\boldsymbol{p}\,\in\,\boldsymbol{\mathcal{P}}} \ \widehat{J}(\boldsymbol{p}) \qquad ; \quad \widehat{J}(\boldsymbol{p}) \quad := \quad \sum_{\text{Angles of illumination}} \frac{1}{2}\ \Big\| \ \Phi(\boldsymbol{p}) \ - \ \mathbf{d}_{\text{obs}} \ \Big\|^2_{\boldsymbol{\mathcal{D}}} \ .} \tag{20}$$

3 Calculation of the gradient by adjoint-state method

We will use a gradient-based optimization, c.f. Section 4, to correct the initial guess to the true models. The main ingredient of these first-order methods is the Fréchet derivatives of the cost function $\widehat{J}$ (versus second-order methods which use the Hessian in addition).

The differentiability of the cost function depends on that of the forward map Φ, c.f. Proposition 1,

$$\partial_{\boldsymbol{p}}\, \widehat{J}(\boldsymbol{p}) \quad = \quad \mathrm{Re}\Big((\Phi - \mathbf{d_{obs}})^{\star}\ \partial_{\boldsymbol{p}}\Phi \Big) \quad .$$

This will entail the differentiability of the scattered field, the multiple-scattering matrix and that of the incident wave (i.e. the RHS). We briefly remind some results on the differentiability of the forward map from the literature. In shape reconstruction, with the range of Φ being far-field patterns, the differentiability of Φ (with respect to the boundary of the obstacles) is given by [26, 27], see also the references cited in [34]. It is noted in [33, p.95 second paragraph], that when the range is the trace of the scattered field on a surface, the differentiability of the forward map (with respect to the boundary of the obstacles) still holds. This fact is also proved in [41] which involves showing the differentiability of layer potentials. The proofs in [41] can be adapted to show the Fréchet differentiability of the single-layer potential and as a result that of the multiple-scattering matrices with respect to the position of arbitrarily shaped obstacles. Since we are working with circular obstacles, we can take advantage of the explicit multipole expansion to show and calculate these derivatives directly. This is even simpler for the discretized problem, since the multi-pole expansions for the scattered field and the scattering matrices comprise only of finite terms. Their explicit calculations are carried out in Appendix B.3.

To calculate $\widehat{J}'$, we can either calculate the Jacobian $\partial_{\boldsymbol{p}}\Phi$ (the sensitivity approach), or only calculate the action of this Jacobian on the vector $\Phi(\boldsymbol{p}) - \mathbf{d}_{\text{obs}}$ via the so-called Adjoint-State method. Option 1, more costly, however provides more information, i.e. the action of the Jacobian on any vector in $\mathbb{C}^N$ and not just the vector $\Phi(\boldsymbol{p}) - \mathbf{d}_{\text{obs}}$. See [30, p.118-119] for discussion of the drawback of this approach. For detailed description of the Adjoint-State method, see [30, 12]. We present the 'traditional' adjoint-state calculation in subsection 3.1 and one via another view point called Adjoint-state by the Lagrangian in subsection 3.2. The results of these calculation are summarized as follows, For a fixed set of parameter p, we calculate $\widehat{J}'(\boldsymbol{p})$, using the following steps.

1. We first solve the direct problem for $V = \mathsf{S}(\boldsymbol{p})$ which is a solution to
$$\mathbf{A}(\boldsymbol{p})\, V \quad = \quad F(\boldsymbol{p}\,,\, u_{\mathrm{pw}}) \quad , \tag{21}$$
where $\mathbf{A}$ is given by (12) and F by (14). After this step, we have also obtained the value of $\widehat{J}$.
2. We then solve the adjoint equation
$$\mathbf{A}(\boldsymbol{p})^{\star}\; \gamma_1 \quad = - \quad \mathfrak{R}^{\star}(\boldsymbol{p})\; \big(\Phi(\boldsymbol{p}) - \mathbf{d}_{\mathrm{obs}}\big)\,, \tag{22}$$
for the adjoint state $\gamma_1 = \boldsymbol{\gamma_1}(\boldsymbol{p})$, which is a vector of size $N = (2\mathbf{m}+1) \times N_{\mathrm{Obs}}$. The matrix $\mathfrak{R}(\boldsymbol{p})$ is defined in (18).
3. The gradient of the reduced cost function $\widehat{J}$ at p is given by , c.f. Prop. 2
$$\widehat{J}'(\boldsymbol{p}) \;=\; \mathrm{Re}\Big[\big(\Phi(\boldsymbol{p}) - \mathbf{d}_{\mathrm{Obs}}\big)^{\star}\; \partial_{\boldsymbol{p}}\mathfrak{R} \underset{(J,l)}{\cdot} \mathsf{S}(\boldsymbol{p}) \;+\; \boldsymbol{\gamma_1}^{\star}\, \Big(\partial_{\boldsymbol{p}}\, \mathbf{A} \underset{(J,l)}{\cdot} \mathsf{S}(\boldsymbol{p}) \;-\; \partial_{\boldsymbol{p}} F(\boldsymbol{p}\,,\, u_{\mathrm{pw}})\Big)\Big]. \tag{23}$$
The quantities and notations in the above expression are explained in Remark 2.

Remark 2.

1. *The first term in the derivative of* $\widehat{J}$ *is related to the derivative of the single layer* $\partial_{\boldsymbol{p}}\mathfrak{R}$ *given in* (66) *in Subsection B.2. The quantity* $\partial_{\boldsymbol{p}}\mathfrak{R} \underset{(J,l)}{\cdot} \mathsf{S}(\boldsymbol{p})$ *is of size* $N_{rec} \times N$*, with its* k*-th row given by the following expression evaluated at* $V = \mathsf{S}(\boldsymbol{p})$*.*
$$\Bigg(\sum_{-\mathbf{m} \le m \le \mathbf{m}} \frac{i\pi\, \mathsf{r}_I}{2}\, \mathrm{J}_m(\kappa\, \mathsf{r}_I) e^{i\, m\, \theta_I(\mathsf{R}_k)}\, V_{Im} \Big[-\kappa_e\, \mathrm{H}_m^{(1)\prime}\big(\kappa\, r_I(\mathsf{R}_k)\big) \begin{pmatrix} \cos\theta_I(\mathsf{R}_k) \\ \sin\theta_I(\mathsf{R}_k) \end{pmatrix}^t + \frac{\mathrm{i}\, m\, \mathrm{H}_m^{(1)}\big(\kappa\, r_I(\mathsf{R}_k)\big)}{\|\mathsf{R}_k - \mathbf{x}_I\|} \begin{pmatrix} \sin\theta_I(\mathsf{R}_k) \\ -\cos\theta_I(\mathsf{R}_k) \end{pmatrix}^t \Big] \Bigg)_{1 \le I \le N_{Obs}},$$
The operation $\underset{(J,l)}{\cdot}$ *means taking the trace with respect to the tuple-indices* $\{(J,l)\}$*.*
2. *The second term in derivative of* $\widehat{J}$ *comes from* $\partial_{\boldsymbol{p}}\, \mathbf{A}$ *calculated in* (59)-(60) *in Subsection B.1. and that for source term associated with planewave,* $\partial_{\boldsymbol{p}} F_{pw}$ *given in Subsection B.3. Its I-th block (in* `Matlab` *syntax) is given by the following expression evaluated at* $W = \overline{\boldsymbol{\gamma_1}}$

and $V = \mathsf{S}(\boldsymbol{p})$.

$$\big(W^t\,\partial_{\boldsymbol{p}}\,\mathbf{A} \underset{(J,l)}{\cdot} V\big)(2I-1:2I) \;=\; \big(\,\partial_{\boldsymbol{p}}\; W^t\mathbf{A}(\boldsymbol{p})\,V\;\big)(2I-1:2I)$$

$$= \sum_{J=1}^{N_{Obs}} \left[\sum_{l,m=-\mathbf{m}}^{\mathbf{m}} W_{Il}\;\boldsymbol{\alpha}_{Il}\;\mathcal{M}^{(1)}_{IJ,lm}\;\boldsymbol{\beta}_{Jm}\;V_{Jm} \;+\; W_{Jm}\;\boldsymbol{\alpha}_{Jm}\;\mathcal{M}^{(2)}_{IJ;lm}\;\boldsymbol{\beta}_{Il}\;V_{Il}\right]\begin{pmatrix}\cos\theta_{JI}\\ \sin\theta_{JI}\end{pmatrix}^t$$

$$+ \sum_{J=1}^{N_{Obs}} \left[\sum_{l,m=-\mathbf{m}}^{\mathbf{m}} W_{Il}\;\boldsymbol{\alpha}_{Il}\;\mathcal{M}^{(3)}_{IJ,lm}\;\boldsymbol{\beta}_{Jm}\;V_{Jm} \;+\; W_{Jm}\;\boldsymbol{\alpha}_{Jm}\;\mathcal{M}^{(4)}_{IJ,lm}\;\boldsymbol{\beta}_{Il}\;V_{Il}\right]\begin{pmatrix}-\sin\theta_{JI}\\ \cos\theta_{JI}\end{pmatrix}^t$$

$$+\,2\,\kappa_e\,\mathbf{u}_{\boldsymbol{inc}}(\mathbf{x}_I)\begin{pmatrix}\cos\alpha_{\boldsymbol{inc}}\\ \sin\alpha_{\boldsymbol{inc}}\end{pmatrix}\sum_{l=-\mathbf{m}}^{\mathbf{m}} W_{Il}\;i^{l+1}\;e^{-il\alpha_{\boldsymbol{inc}}}\times\begin{cases}\mathrm{J}_l(\kappa_e\mathsf{r}_I) & \textit{Dir BC}\\ \kappa\mathrm{J}'_l(\kappa\,\mathsf{r}_I) & \textit{Neu BC}\end{cases}.$$

The block matrix $\mathcal{M}^{(1)}_{IJ}$, $\mathcal{M}^{(2)}_{IJ}$, $\mathcal{M}^{(3)}_{IJ}$, $\mathcal{M}^{(4)}_{IJ}$, *and matrix* α *and* β *are defined in* (62) *and* (61).

Remark 3. *In practice, to take advantage of the feature of Mumps and Lapack which allows users to solve also the transpose problem, by re-using the factorization already made in step one (for solving the direct equation* (21)*), instead of working with* (22)*, we solve for* $\overline{\gamma_1}$ *the unique solution to*

$$\mathbf{A}(\boldsymbol{p})^t\;\overline{\gamma_1} \quad = - \quad \big(\mathfrak{R}(\boldsymbol{p})\big)^t\;\overline{\Phi(\boldsymbol{p}) - \mathbf{d}_{obs}} \quad .$$

3.1 The adjoint-state method

Preliminary step We write the derivatives of $\widehat{J}$ in terms of that Φ. Denote by Φ_i the real and imaginary part of Φ. Since $\boldsymbol{p}$ is a real variable, the Jacobian and Fréchet derivatives of Φ and $\widehat{J}$ are well-defined, under sufficient regularity assumptions.

$$\Phi(\boldsymbol{p}) \;=\; \Phi_1(\boldsymbol{p}) \;+\; \mathrm{i}\,\Phi_2(\boldsymbol{p}) \quad;\quad \Phi_i : \mathbb{R}^{N_{\text{par}}} \to \mathbb{R}^{N_{\text{rec}}} \quad .$$

If exists, $\partial_{\boldsymbol{p}}\Phi_i$ is a matrix of size $N_{\text{rec}} \times N_{\text{par}}$. The reduced cost function $\widehat{J}$ is a real valued function of a real variable , $\widehat{J} : \mathbb{R}^{N_{\text{par}}} \to \mathbb{R}$,

$$\widehat{J}(\boldsymbol{p}) \;=\; \tfrac{1}{2}\|\Phi_1(\boldsymbol{p}) - \mathbf{d}_1\|^2 \;+\; \tfrac{1}{2}\|\Phi_2(\boldsymbol{p}) - \mathbf{d}_2\|^2 \quad .$$

Thus its gradient ∇ and derivative can be defined as normal.

Proposition 1. *We have*

$$\begin{aligned}\underbrace{\partial_{\boldsymbol{p}}\,\widehat{J}}_{1\times N_{par}} \;&=\; \underbrace{\Big(\Phi_1(\boldsymbol{p}) - \mathbf{d}_1\Big)^t}_{1\times N_{rec}}\;\underbrace{\partial_{\boldsymbol{p}}\Phi_1}_{N_{rec}\times N_{par}} \;+\; \Big(\Phi_2(\boldsymbol{p}) - \mathbf{d}_2\Big)^t\;\partial_{\boldsymbol{p}}\Phi_2\\ &=\; \mathrm{Re}\Big[\;\Big(\Phi - \mathbf{d}_{\boldsymbol{obs}}\Big)^{\star}\;\partial_{\boldsymbol{p}}\Phi\;\Big]\end{aligned} \tag{24}$$

In the above expressions, the involved operations are matrix vector product.

We also write

$$\nabla_{\boldsymbol{p}}\,\widehat{J}(\boldsymbol{p}) \quad=\quad \mathrm{Re}\Big[\Big(\partial_{\boldsymbol{p}}\Phi\Big)^{\star}\;\Big(\Phi - \mathbf{d}_{\mathbf{obs}}\Big)\Big] \quad . \tag{25}$$

Proof. Write

$$\mathbb{C}^{N_{\rm obs}} \ni \mathbf{d}_{\rm obs} \;=\; \mathbf{d}_1 \;+\; \mathrm{i}\,\mathbf{d}_2 \quad , \quad \mathbf{d}_i \in \mathbb{R}^{N_{\rm rec}} \quad .$$

For $\delta_{\boldsymbol{p}} \in \mathbb{R}^{N_{\rm par}}$,

$$\begin{aligned}
\Big\langle \partial_{\boldsymbol{p}}\, \widehat{\mathcal{J}} \;,\; \delta_{\boldsymbol{p}} \Big\rangle_{(\mathbb{R}^{N_{\rm par}})^\star \times \mathbb{R}^{N_{\rm par}}} \;&=\; \Big\langle \partial_{\boldsymbol{p}}\, \tfrac{1}{2}\, \| \Phi_1(\boldsymbol{p}) - \mathbf{d}_1 \|^2 \;+\; \partial_{\boldsymbol{p}}\, \tfrac{1}{2}\, \| \Phi_2(\boldsymbol{p}) - \mathbf{d}_2 \|^2 \;,\; \delta_{\boldsymbol{p}} \Big\rangle_{(\mathbb{R}^{N_{\rm par}})^\star \times \mathbb{R}^{N_{\rm par}}} \\
&=\; \Big((\partial_{\boldsymbol{p}} \Phi_1)\, \delta_{\boldsymbol{p}} \;,\; \Phi_1(\boldsymbol{p}) - \mathbf{d}_1 \Big) \;+\; \Big((\partial_{\boldsymbol{p}} \Phi_2)\, \delta_{\boldsymbol{p}} \,,\, \Phi_2(\boldsymbol{p}) - \mathbf{d}_2) \Big) \\
&=\; \Big(\delta_{\boldsymbol{p}} \;,\; (\partial_{\boldsymbol{p}} \Phi_1)^t \,\big(\Phi_1(\boldsymbol{p}) - \mathbf{d}_1\big) \Big) \;+\; \Big(\;\delta_{\boldsymbol{p}} \;,\; (\partial_{\boldsymbol{p}} \Phi_2)^t \,\big(\Phi_2(\boldsymbol{p}) - \mathbf{d}_2\big) \Big) \\
&=\; \Big[\big(\Phi_1(\boldsymbol{p}) - \mathbf{d}_1\big)^t\, \partial_{\boldsymbol{p}} \Phi_1 \;+\; \big(\Phi_2(\boldsymbol{p}) - \mathbf{d}_2\big)^t\, \partial_{\boldsymbol{p}} \Phi_2 \;\Big]\, \delta_{\boldsymbol{p}} \quad .
\end{aligned}$$

Here (,) represents the vector dot product in $\mathbb{R}^{N_{\rm par}}$. Thus

$$\partial_{\boldsymbol{p}}\, \widehat{\mathcal{J}} \quad=\quad \big(\Phi_1(\boldsymbol{p}) - \mathbf{d}_1\big)^t\, \partial_{\boldsymbol{p}} \Phi_1 \;+\; \big(\Phi_2(\boldsymbol{p}) - \mathbf{d}_2\big)^t\, \partial_{\boldsymbol{p}} \Phi_2 \quad .$$

We calculate the real and imaginary part of $(\Phi - \mathbf{d_{obs}})^\star\; \partial_{\boldsymbol{p}} \Phi$,

$$\begin{aligned}
&(\Phi - \mathbf{d_{obs}})^\star\; \partial_{\boldsymbol{p}} \Phi \\
&= \Big[\big(\Phi_1 - \mathbf{d}_1\big) + \mathrm{i}\big(\Phi_2 - \mathbf{d}_2\big) \Big]^\star\; \Big(\partial_{\boldsymbol{p}} \Phi_1 \;+\; \mathrm{i}\, \partial_{\boldsymbol{p}} \Phi_2 \Big) \\
&= \Big[\big(\Phi_1 - \mathbf{d}_1\big)^t - \mathrm{i}\big(\Phi_2 - \mathbf{d}_2\big)^t \Big]\; \Big(\partial_{\boldsymbol{p}} \Phi_1 \;+\; \mathrm{i}\, \partial_{\boldsymbol{p}} \Phi_2 \Big) \\
&= \big(\Phi_1 - \mathbf{d}_1\big)^t\, \partial_{\boldsymbol{p}} \Phi_1 \;+\; \big(\Phi_2 - \mathbf{d}_2\big)^t\, \partial_{\boldsymbol{p}} \Phi_2 \;+\; \mathrm{i}\, \Big[\big(\Phi_1 - \mathbf{d}_1\big)^t\, \partial_{\boldsymbol{p}} \Phi_2 \;-\; \big(\Phi_2 - \mathbf{d}_2\big)^t \partial_{\boldsymbol{p}}\, \Phi_1 \Big] \quad .
\end{aligned}$$

As a result,

$$\partial_{\boldsymbol{p}}\, \widehat{\mathcal{J}}(\boldsymbol{p}) \quad=\quad \mathrm{Re}\, \Big((\Phi - \mathbf{d_{obs}})^\star\; \partial_{\boldsymbol{p}} \Phi \Big) \quad .$$

□

Step 1 We further explore the structure of the Jacobian $\partial_{\boldsymbol{p}} \Phi$, with a goal to see its specific action on the vector $\Phi(\boldsymbol{p}) - \mathbf{d}_{\rm obs}$.

$$\underbrace{\Phi(\boldsymbol{p})}_{\substack{\text{of size} \\ N_{\rm rec} \times 1}} \;=\; \underbrace{\boldsymbol{\mathfrak{R}}(\boldsymbol{p})}_{\substack{\text{of size} \\ N_{\rm rec} \times N}} \;\; \underbrace{\mathsf{S}(\boldsymbol{p})}_{\substack{\text{col. vector} \\ \text{of size } N}} \;=\; \left(\sum_{J=1}^{N_{\rm Obs}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} (\mathcal{S}_J\, \mathbf{w}_l)(\mathsf{R}_k)\; \mathsf{S}_{Jl} \right)_{1 \le k \le N_{\rm rec}}$$

Recall that $\boldsymbol{\mathfrak{R}}$ is matrix of size $N_{\rm rec} \times N$, with row indexed by $1 \le k \le N_{\rm rec}$ and columns indexed by couple (J, l) with $1 \le J \le N_{\rm obs}$, $-\mathbf{m} \le l \le \mathbf{m}$.

$$\underbrace{\partial_{\boldsymbol{p}}\; \Phi}_{\text{of size } N_{\rm rec} \times N_{\rm par}} \;=\; \left(\underbrace{\partial_{\boldsymbol{p}} \sum_{J=1}^{N_{\rm Obs}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} (\mathcal{S}_J\, \mathbf{w}_l)(\mathsf{R}_k)\; \mathsf{S}_{Jl}}_{\text{row vector of size } N_{\rm par}} \right)_{1 \le k \le N_{\rm rec}}$$

At each index k, the quantity in the parentheses is a row vector of size $N_{\text{par}} = 2N_{\text{Obs}}$. We calculate the components of this row vector.

$$
\begin{aligned}
\underbrace{\partial_{\boldsymbol{p}}\,\Phi}_{\text{of size } N_{\text{rec}}\times N_{\text{par}}} \;&=\; \left(\partial_{\boldsymbol{p}} \sum_{J=1}^{N_{\text{Obs}}}\sum_{l=-\mathbf{m}}^{\mathbf{m}} (\mathcal{S}_J\,\mathbf{w}_l)(\mathsf{R}_k)\;\mathsf{S}_{Jl}\right)_{1\le k\le N_{\text{rec}}} \\
= \left(\sum_{J=1}^{N_{\text{Obs}}}\sum_{l=-\mathbf{m}}^{\mathbf{m}} \Big(\mathcal{S}_J\,\mathbf{w}_l\Big)(\mathsf{R}_k)\,\Big[\partial_{\boldsymbol{p}}\,\mathsf{S}_{Jl}\Big]\right)_{1\le k\le N_{\text{rec}}} &+ \left(\sum_{J=1}^{N_{\text{Obs}}}\sum_{l=-\mathbf{m}}^{\mathbf{m}} \Big[\partial_{\boldsymbol{p}}\,(\mathcal{S}_J\,\mathbf{w}_l)(\mathsf{R}_k)\Big]\;\mathsf{S}_{Jl}\right)_{1\le k\le N_{\text{rec}}} \\
= \;\; \mathfrak{R}\,\partial_{\boldsymbol{p}}\,\mathsf{S} \;+\; (\partial_{\boldsymbol{p}}\,\mathfrak{R}) \underset{(J,l)}{\cdot} \mathsf{S} \quad . &
\end{aligned}
\tag{26}
$$

The row of the second matrix in (26) is calculated in (66), with the expression evaluated at $V = \mathsf{S}(\boldsymbol{p})$.

It remains to study the first term in (26). we write

$$
\begin{aligned}
& \mathbf{A}(\boldsymbol{p})\,\mathsf{S}(\boldsymbol{p}) && = \;\; F(\boldsymbol{p}) \\
\Rightarrow \quad & \mathbf{W}(\boldsymbol{p}) \;\;+\;\; \mathbf{A}(\boldsymbol{p})\;\partial_{\boldsymbol{p}}\,\big[\mathsf{S}(\boldsymbol{p})\big] && = \;\; \partial_{\boldsymbol{p}} F(\boldsymbol{p}) \quad .
\end{aligned}
$$

Here, $\mathbf{W}(\boldsymbol{p})$ is matrix of size $N_{\mathbf{sol}} \times N_{\mathbf{par}}$,

$$
\mathbf{W} \;=\; \partial_{\boldsymbol{p}}\mathbf{A} \underset{(J,l)}{\cdot} \mathsf{S}
\tag{27}
$$

with its (I,l) row given by

$$
\mathbf{W}((I,l),:) \;=\; \left(\underbrace{\sum_{\tilde{J}=1}^{N_{\mathbf{Obs}}}\sum_{m=-\mathbf{m}}^{\mathbf{m}} \big(\partial_{\boldsymbol{p}}\,\mathbf{A}_{I\tilde{J};lm}\big)\,\mathsf{S}_{\tilde{J}m}}_{\text{the } (I,l)\text{-th row of } W}\right)_{\substack{1\le I\le N_{\mathbf{Obs}} \\ -\mathbf{m}\le l\le \mathbf{m}}} \quad .
\tag{28}
$$

As a result, we obtain

$$
\partial_{\boldsymbol{p}}\,\mathsf{S}(\boldsymbol{p}) \quad = \quad \mathbf{A}(\boldsymbol{p})^{-1}\,\Big(-\,\boldsymbol{W}(\boldsymbol{p}) \;+\; \partial_{\boldsymbol{p}}\,F(\boldsymbol{p})\Big) \quad .
\tag{29}
$$

Step 2 We substitute from (26) the calculated expression of $\partial_{\boldsymbol{p}}\Phi$.

$$
(\Phi - \mathbf{d_{obs}})^{\star}\;\partial_{\boldsymbol{p}}\Phi \;=\; (\Phi - \mathbf{d_{obs}})^{\star}\,\mathfrak{R}\,\partial_{\boldsymbol{p}}\mathsf{S} \;+\; (\Phi - \mathbf{d_{obs}})^{\star}\;(\partial_{\boldsymbol{p}}\,\mathfrak{R}) \underset{(J,l)}{\cdot} \mathsf{S}.
\tag{30}
$$

We rewrite further the **first term** by substituting in (29) (by convention, all vectors are column),

$$
\begin{aligned}
(\Phi - \mathbf{d_{obs}})^{\star}\,\mathfrak{R}\,\partial_{\boldsymbol{p}}\mathsf{S} &= \underbrace{\Big(\Phi - \mathbf{d_{obs}}\Big)^{\star}\,\mathfrak{R}}_{\text{of size } 1\times N_{\mathbf{sol}}} \quad \underbrace{\mathbf{A}^{-1}\Big(-\mathbf{W} + \partial_{\boldsymbol{p}}F\Big)}_{N_{\mathbf{sol}}\times N_{\mathbf{par}}} \\
&= \Big(\mathfrak{R}^{\star}\,(\Phi - \mathbf{d_{obs}})\Big)^{\star}\;\mathbf{A}^{-1}\Big(-\mathbf{W} + \partial_{\boldsymbol{p}}F\Big) \\
&= \Big(\mathbf{A}^{-\star}\,\mathfrak{R}^{\star}\,(\Phi - \mathbf{d_{obs}})\Big)^{\star}\;\Big(-\mathbf{W} + \partial_{\boldsymbol{p}}F\Big) \\
&= \boldsymbol{\gamma}_{\mathbf{1}}^{\star}\;\Big(-\mathbf{W} + \partial_{\boldsymbol{p}}F\Big)\,.
\end{aligned}
$$

We have denoted the expression in the first parenthesis by $\boldsymbol{\gamma_1}$,

$$\begin{aligned} \boldsymbol{\gamma_1} \quad &= \quad -\mathbf{A}^{-\star}\,\mathfrak{R}^{\star}\Big(\Phi - \mathbf{d_{obs}}\Big) \quad ; \\ \Leftrightarrow \; \mathbf{A}^{\star}\,\boldsymbol{\gamma_1} \quad &= \quad -\mathfrak{R}^{\star}\Big(\Phi - \mathbf{d_{obs}}\Big) \quad . \end{aligned}$$

Now using Prop. 1, we arrive at an expression to calculate derivative of $\widehat{J}$.

Proposition 2.

$$\begin{aligned} \widehat{J}'(\boldsymbol{p}) &= \mathrm{Re}\Big[\big(\mathfrak{R}V - \mathbf{d}_{\boldsymbol{obs}}\big)^{\star}\;\partial_p(\mathfrak{R}\,V) \;+\; \gamma_1^{\star}\,\big(\partial_{\boldsymbol{p}}\,(\mathbf{A}\,V) \;-\; \partial_{\boldsymbol{p}}F\big)\Big]\Big|_{V=\mathsf{S}(\boldsymbol{p})\,,\,\gamma_1=\boldsymbol{\gamma_1}} \\ &= \mathrm{Re}\Big[\big(\Phi(\boldsymbol{p}) - \mathbf{d}_{\boldsymbol{obs}}\big)^{\star}\;\partial_p\mathfrak{R}\underset{(J,l)}{\cdot}\mathsf{S}(\boldsymbol{p}) \;+\; \boldsymbol{\gamma_1^{\star}}\,\Big(\partial_{\boldsymbol{p}}\,\mathbf{A}\underset{(J,l)}{\cdot}\mathsf{S}(\boldsymbol{p}) \;-\; \partial_{\boldsymbol{p}}F\Big)\Big] \qquad (31) \\ &= \mathrm{Re}\Big[\big(\Phi(\boldsymbol{p}) - \mathbf{d}_{\boldsymbol{obs}}\big)^{\star}\;\partial_p\mathfrak{R}\underset{(J,l)}{\cdot}\mathsf{S}(\boldsymbol{p}) \;+\; \boldsymbol{\gamma_1^{\star}}\,\big(\boldsymbol{\mathsf{W}} \;-\; \partial_{\boldsymbol{p}}F\big)\Big]\,. \end{aligned}$$

3.2 The adjoint-state method by the Lagrangian

We now re-derive formula (23) from the Lagrangian point of view. We have introduced the function spaces,

- $\boldsymbol{\mathcal{P}} \subset \mathbb{R}^{N_{\text{par}}}$ is the space of real parameters we aim to retrieve.
- $\boldsymbol{\mathcal{D}} \subset \mathbb{C}^{N_{\text{rec}}}$ represents the space of observed data at these receivers.
- $\boldsymbol{\mathcal{V}} = \mathbb{C}^{N}$ is a space of solution of the multiple-scattering linear system.

Introduce the cost function $\mathcal{G}$ defined as

$$\begin{aligned} \mathcal{G} \;:\; \mathbf{P} \times \boldsymbol{\mathcal{V}} \quad &\longrightarrow \quad \mathbb{R} \\ \mathcal{G}(\boldsymbol{p}, V) \;:=\; &\Big\| \mathfrak{R}(\boldsymbol{p})\;V \;-\; \mathbf{d}_{\text{obs}} \Big\|^2_{\boldsymbol{\mathcal{D}}}. \end{aligned} \qquad (32)$$

We introduce the **state operator**

$$S : \mathbf{P} \times \boldsymbol{\mathcal{V}} \to \mathbb{C}^N \qquad S(\boldsymbol{p}, V) \;=\; \mathbf{A}(\boldsymbol{p})\,V - F(\boldsymbol{p}) \quad ; \qquad (33)$$

and **state equation**

$$S(\boldsymbol{p}, V) \;=\; 0\;. \qquad (34)$$

By the definition of the solution operator given in (11), we have

$$S(\boldsymbol{p}\;,\;\mathsf{S}(\boldsymbol{p})) \;=\; 0$$

In addition, by the definition of the forward map Φ in (17), the cost function $\mathcal{G}$ evaluated at $V = \mathsf{S}(\boldsymbol{p})$ gives the reduced cost function $\widehat{J}$,

$$\widehat{J}(\boldsymbol{p}) \quad := \quad \mathcal{G}\Big(\boldsymbol{p}\;,\;\mathsf{S}(\boldsymbol{p})\Big) \;=\; \Big\|\Phi(\boldsymbol{p}) \;-\; \mathbf{d}_{\text{obs}}\Big\|^2_{\boldsymbol{\mathcal{D}}}. \qquad (35)$$

Lagrangian To calculate $\widehat{J}'$, we consider the following optimization problem

$$\min_{\substack{\boldsymbol{p}\,\in\,\mathbf{P}\\ V\in\boldsymbol{\mathcal{V}}}} \;\mathcal{G}(\boldsymbol{p},V) \quad \text{subject to} \quad S\,(\,\boldsymbol{p}\,,\,V)\;=\;0\,,$$

with $\boldsymbol{p}$ and V treated as independent variables, with their relation given by state equation (34) posed as a constraint. For this purpose, we introduce the Lagrangian $\mathcal{L}$ and the adjoint state variable $\gamma_1 \in \boldsymbol{\mathcal{V}} = \mathbb{C}^N$

$$\begin{array}{rcl} \mathcal{L} \;:\; \boldsymbol{\mathcal{P}} \times \boldsymbol{\mathcal{V}} \times \boldsymbol{\mathcal{V}}^\star & \longrightarrow & \mathbb{C} \\ \mathcal{L}(\boldsymbol{p},V,\gamma_1) \;:=\; \mathcal{G}(\boldsymbol{p},V) & + & \big\langle \gamma_1\,,\, S(\boldsymbol{p},V)\big\rangle_{\boldsymbol{\mathcal{V}}^\star,\boldsymbol{\mathcal{V}}} \quad . \end{array} \tag{36}$$

Since there is a natural identification of $\boldsymbol{\mathcal{V}}^\star$ with $\boldsymbol{\mathcal{V}}$, we will simply write the second term in (36) using matrix-vector product,

$$\mathfrak{l}(\boldsymbol{p},V,\gamma_1) \;:=\; \big\langle \gamma_1 \;,\; S(\boldsymbol{p},V)\big\rangle_{\boldsymbol{\mathcal{V}}^\star,\boldsymbol{\mathcal{V}}} \;=\; \gamma_1^\star \; S(\boldsymbol{p}\,,\,V)\;.$$

Here * is the transposed complex conjugate.

We have

$$\widehat{J}(\boldsymbol{p}) \;=\; \text{Re}\;\,\mathcal{L}\big(\boldsymbol{p}\,,\,\mathsf{S}(\boldsymbol{p})\,,\,\gamma_1\big) \quad , \quad \forall\,\gamma_1\,,\,\boldsymbol{p}.$$

And,

$$\widehat{J}'(\boldsymbol{p}) \;=\; \partial_{\boldsymbol{p}}\,\Big[\text{Re}\;\,\mathcal{L}\big(\boldsymbol{p}\,,\,\mathsf{S}(\boldsymbol{p})\,,\,\gamma_1\big)\Big]\,.$$

By (56) in Prop 4 of Appendix A,

$$\widehat{J}'(\boldsymbol{p})\,\delta_{\boldsymbol{p}} \;=\; \text{Re}\,\Big[\big(\partial_{\boldsymbol{p}}\mathcal{L}\big)(\boldsymbol{p},V,\gamma_1) + \big(\partial_V^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p},V,\gamma_1)\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p}) + \big(\partial_{\overline{V}}^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p},V,\gamma_1)\;\partial_{\boldsymbol{p}}\,\overline{\mathsf{S}(\boldsymbol{p})}\Big]\Big|_{V=\mathsf{S}(\boldsymbol{p})}\,\delta_{\boldsymbol{p}}\,. \tag{37}$$

Here, $\partial_{\boldsymbol{p}}\mathcal{L}$ is the usual partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{p}$, while $\partial_V^{\mathsf{CR}}\mathcal{L}$, and $\partial_{\overline{V}}^{\mathsf{CR}}\mathcal{L}$ are the complex partial derivatives defined in Definition 1 of Appendix A.

The adjoint state problem We would like to avoid calculating $\partial_{\boldsymbol{p}}\,\overline{\mathsf{S}(\boldsymbol{p})}$. For this purpose, we can and will choose γ_1 so that the terms in (37) containing this quantity and its conjugate vanish, i.e.

$$\big(\partial_V^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p},V,\boldsymbol{\gamma_1})\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p}) \;+\; \big(\partial_{\overline{V}}^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p}\,,\,V\,,\,\boldsymbol{\gamma_1})\;\partial_{\boldsymbol{p}}\overline{\mathsf{S}(\boldsymbol{p})} \;=\; 0\,. \tag{38}$$

To arrive at a more explicit equation, we now calculate $\mathcal{L}_V$, and $\mathcal{L}_{\overline{V}}$. We have

$$\partial_V^{\mathsf{CR}}\,\mathcal{L} \;=\; \partial_V^{\mathsf{CR}}\,\mathcal{G} \;+\; \partial_V^{\mathsf{CR}}\,\mathfrak{l} \quad ; \quad \partial_{\overline{V}}^{\mathsf{CR}}\,\mathcal{L} \;=\; \partial_{\overline{V}}^{\mathsf{CR}}\,\mathcal{G} \;+\; \partial_{\overline{V}}^{\mathsf{CR}}\,\mathfrak{l}\,.$$

By its definition, $\mathfrak{l}$ is linear in S which is itself linear in V, thus $\mathfrak{l}$ is Fréchet differentiable with respect to V and

$$\partial_V^{\mathsf{CR}}\,\mathfrak{l} \;=\; \partial_V^{\mathsf{CR}}\,\Big[\boldsymbol{\gamma_1^\star}(\mathbf{A}V - F)\Big] \;=\; \boldsymbol{\gamma_1^\star}\mathbf{A} \quad ; \quad \partial_{\overline{V}}^{\mathsf{CR}}\,\mathfrak{l} \;=\; 0\,.$$

By Definition 1 in Appendix A, we have

$$\partial_V^{\mathsf{CR}}\,\mathcal{G} \;=\; \tfrac{1}{2}(\mathfrak{R}\;V - \mathbf{d_{obs}})^\star\,\mathfrak{R} \quad ; \quad \partial_{\overline{V}}^{\mathsf{CR}}\,\mathcal{G} \;=\; \tfrac{1}{2}(\mathfrak{R}\,V - \mathbf{d_{obs}})^t\,\overline{\mathfrak{R}}\,.$$

LHS of condition (38) is

$$\begin{aligned}
&\big(\partial_V^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p}\,,\,V\,,\,\gamma_1)\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p}) \;+\; \big(\partial_{\overline{V}}^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p}\,,\,V\,,\,\gamma_1)\;\partial_{\boldsymbol{p}}\overline{\mathsf{S}(\boldsymbol{p})}\\
&= \tfrac{1}{2}\Big(\mathfrak{R}V - \mathbf{d_{obs}}\Big)^{\star}\;\mathfrak{R}\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p}) \;+\; \tfrac{1}{2}\Big(\mathfrak{R}\,V - \mathbf{d_{obs}}\Big)^{t}\;\overline{\mathfrak{R}}\;\partial_{\boldsymbol{p}}\overline{\mathsf{S}(\boldsymbol{p})} \;+\; \gamma_1^{\star}\mathbf{A}\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p})\\
&= \;\mathrm{Re}\Big[(\mathfrak{R}\,V - \mathbf{d_{obs}})^{\star}\;\mathfrak{R}\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p})\Big] \;+\; \gamma_1^{\star}\,\mathbf{A}\,\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p})\,.
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{Re}\Big[\big(\partial_V^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p},V,\gamma_1)\;\partial_{\boldsymbol{p}}\mathsf{S}(\boldsymbol{p}) \;+\; \big(\partial_{\overline{V}}^{\mathsf{CR}}\mathcal{L}\big)(\boldsymbol{p},V,\gamma_1)\;\partial_{\boldsymbol{p}}\overline{\mathsf{S}(\boldsymbol{p})}\Big]\\
&\quad=\;\mathrm{Re}\Big[\Big(\big(\mathfrak{R}V - \mathbf{d_{obs}}\big)^{\star}\;\mathfrak{R}\;+\;\gamma_1^{\star}\,\mathbf{A}\Big)\;\partial_{\boldsymbol{p}}\;\mathsf{S}(\boldsymbol{p})\Big]\,.
\end{aligned}$$

Condition (38) is satisfied if γ_1 solves

$$\begin{aligned}
&\big(\mathfrak{R}V - \mathbf{d_{obs}}\big)^{\star}\,\mathfrak{R} \;+\; \gamma_1^{\star}\,\mathbf{A} \;=\; 0 \quad , \quad V = \mathsf{S}(\boldsymbol{p})\\
\Rightarrow\quad &\mathbf{A}^{\star}\gamma_1 \;=\; -\mathfrak{R}^{\star}\;\big(\mathfrak{R}V - \mathbf{d_{obs}}\big) \quad , \quad V = \mathsf{S}(\boldsymbol{p})\,.
\end{aligned}$$

We arrive at the adjoint problem with solution $\boldsymbol{\gamma_1}$,

$$\boxed{\mathbf{A}^{\star}\,\gamma_1 \;=\; -\mathfrak{R}^{\star}\;\Big(\Phi(\boldsymbol{p}) - \mathbf{d_{obs}}\Big)\;.} \tag{39}$$

Final step We use the state adjoint $\gamma_1 = \boldsymbol{\gamma_1}$ as defined in (39) to simplify (37) to

$$\hat{J}'(\boldsymbol{p})\;\delta_{\boldsymbol{p}} \;=\; \mathrm{Re}\Big[\big(\partial_{\boldsymbol{p}}\mathcal{L}\big)\;(\boldsymbol{p},V,\gamma_1)\,\Big|_{V=\mathcal{S}(\boldsymbol{p})\,,\,\gamma_1=\boldsymbol{\gamma_1}}\quad\delta_{\boldsymbol{p}}\Big]\,.$$

It remains to calculate $\partial_{\boldsymbol{p}}\mathcal{L}$

$$\big(\partial_{\boldsymbol{p}}\mathcal{L}\big)(\boldsymbol{p},V,\gamma_1) \;=\; \big(\partial_{\boldsymbol{p}}\mathcal{G}\big)(\boldsymbol{p},V,\gamma_1) \;+\; \big(\partial_{\boldsymbol{p}}\,\mathfrak{l}\big)(\boldsymbol{p},V,\gamma_1)\,.$$

Upon identifying $\mathcal{G}$ with function $\mathcal{H}$ of complex variable $\mathbf{d}$ and $\overline{\mathbf{d}}$,

$$\mathcal{G}(\boldsymbol{p},V) \;=\; \mathcal{H}\big(\mathfrak{R}(\boldsymbol{p})\,V\,,\,\overline{\mathfrak{R}(\boldsymbol{p})\,V}\big)$$

with

$$\mathcal{H}\;:\;\mathbb{C}^{N_{\mathrm{rec}}}\times\mathbb{C}^{N_{\mathrm{rec}}}\;\longrightarrow\;\mathbb{R}\quad;\quad\mathcal{H}(\mathbf{d},\overline{\mathbf{d}})\;:=\;\tfrac{1}{2}\|\mathbf{d}-\mathbf{d}_{\mathrm{obs}}\|^2\,.$$

Note that V is treated as a independent of $\boldsymbol{p}$. Using (55) in Prop 4 of Appendix A, we obtain

$$\begin{aligned}
\big(\partial_{\boldsymbol{p}}\mathcal{G}\big)(\boldsymbol{p},V,\gamma_1) \;&=\; 2\,\mathrm{Re}\Big[(\partial_{\mathbf{d}}^{\mathsf{CR}}\mathcal{H})\;(\partial_{\boldsymbol{p}}\,\mathbf{d})\Big]\,\Big|_{\mathbf{d}\,=\,\mathfrak{R}(\boldsymbol{p})\,V}\\
&=\; 2\,\mathrm{Re}\Big[\tfrac{1}{2}\big(\mathfrak{R}V - \mathbf{d_{obs}}\big)^{\star}\;\partial_{\boldsymbol{p}}(\mathfrak{R}\,V)\Big]\,.
\end{aligned}$$

On the other hand,

$$\big(\partial_{\boldsymbol{p}}\mathfrak{l}\big)(\boldsymbol{p},V,\gamma_1) \;=\; \gamma_1^{\star}\;\big[\partial_{\boldsymbol{p}}(\mathbf{A}V) \;-\; \partial_{\boldsymbol{p}}F\big]\,.$$

Note that, in the above expression, V is a variable independent of p. As a result,

$$\boxed{\hat{J}'(\boldsymbol{p}) \;=\; \mathrm{Re}\Big[\big(\mathfrak{R}V - \mathbf{d_{obs}}\big)^{\star}\;\partial_{\boldsymbol{p}}(\mathfrak{R}\,V) \;+\; \gamma_1^{\star}\Big[\partial_{\boldsymbol{p}}\,(\mathbf{A}\,V) \;-\; \partial_{\boldsymbol{p}}F\Big]\Big|_{V=\mathsf{S}(\boldsymbol{p})\,,\,\gamma_1=\boldsymbol{\gamma_1}}}\,. \tag{40}$$

4 Optimization methods

In this section, we describe the algorithm to correct a prescribed initial guess to the true model. Using multi-frequency data, this process, called frequency-hopping, comprises of running several gradient-based optimization procedures over different frequencies, one for each frequency and from low to high. Each step to update or correct a guess within a gradient-based procedure at a fixed frequency uses linesearch-based strategy, and is described in subsection 4.1. To see how this fits within the final procedure, see Figure 2. The overall frequency-hopping is described in subsection 4.2. General description of linesearch and gradient-based optimization can be found in [40].

4.1 Line search algorithm

We now describe how to go from the current iterate $\boldsymbol{p}_k$ to a next value $\boldsymbol{p}_{k+1}$ at which the (cost) function $\widehat{J}$ has a lower value. The main ingredients of line-search based strategies (opposed to the trust region strategy) for an update are a search direction $\mathbf{s}_k$ (usually chosen to be a descent one) and a step length α_k in this direction, i.e.

$$\boldsymbol{p}_{k+1} \;=\; \boldsymbol{p}_k \,+\, \alpha_k\,\mathbf{s}_k$$

That direction $\mathbf{s}_k$ is a descent direction means

$$\mathbf{s}_k^t\;\nabla\widehat{J}_k < 0 \quad , \quad \nabla\widehat{J}_k \;:=\; \nabla\widehat{J}(\boldsymbol{p}_k)\,.$$

Step length α_k can be found by solving the one-dimensional minimization problem

$$\min_{\alpha>0}\quad \widehat{J}\big(\boldsymbol{p}_k \;+\; \alpha\,\mathbf{s}_k\big)\,. \tag{41}$$

Since exact minimization can be expensive, one uses a **line-search algorithm** to generate a limited number of trial step lengths until one arrives at a candidate that approximates the minimum of (41).

In our numerical experiments, we investigate the effectiveness of two search directions Broyen-Fletcher-Goldfarb-Shannon (BFGS) in the quasi-Newton (QN) family, and Polak-Ribière (PR) with restart in the nonlinear-conjugate gradient (NLCG). These two search directions will be used in combination with the following three line-search algorithms which will be described in details below. The linesearch algorithm we consider are: Linesearch (LS) Algo. 1 also called simple-backtracking is the cheapest, LS Algo. 2 imposes sufficient descent and second-order quadrature for update, and the most expensive one is strong Wolfe LS Algo. 3.

Write

$$\phi_k(\alpha) := \widehat{J}(\boldsymbol{p}_k + \alpha\,\mathbf{s}_k)\ ;\ \phi_k(0) := \widehat{J}(\boldsymbol{p}_k)\ ;\ \phi_k'(0) = \widehat{J}'(\boldsymbol{p}_k)\mathbf{s}_k\,.$$

Linesearch Algo 1 is described in Algo. 1. This is the ‘cheapest’ linesearch requiring only $\phi(\alpha) < \phi(0)$, and imposing no control over how much $\widehat{J}$ decreases. Thus, only the value of the cost function at each trial step size is required. The initial steplength in Algo. 1 is given by,

$$\beta_{\text{initial trial}} \;=\; \frac{\substack{\text{Reference}\\\text{steplength}}}{\|\mathbf{s}_k\|_2/\|\mathbf{s}_k\|_\infty}\,.$$

Linesearch Algo 2 is described in Algo. 2. Here, the degree of decrease in $\widehat{J}$ is imposed by

$$\phi(\alpha) \;\le\; \phi(0) \;+\; \mu\;\alpha\;\phi'(0) \quad , \quad \mu\in(0,1) \qquad \textbf{Armijo condition}\;.$$

```
Set $\beta_{\text{trial}} = \beta_{\text{initial trial}}$;
Set $i = 0$ ;
while $(i \; < \; N_{LS\ IterMax\ 1})$ do
    if $\phi(\beta_{trial}) \; < \; \phi(0)$ then
        $\beta_\star = \beta_{\text{trial}}$ ; LS = ‘successful’ ; Exit ;
    else if $i = N_{LS\ IterMax\ 1} - 1$ then
        $\beta_\star = \beta_{\text{trial}}$ ; LS = ‘unsuccessful’ ; Exit;
    else
        $\beta\text{trial} \leftarrow \frac{1}{2} \times \beta_{\text{trial}}$ ; $i \leftarrow i+1$ ;          /* Update $\beta_{\text{trial}}$ */
    end
end
Result: Set $\alpha_k = \beta_\star$
```

Algorithm 1: Simple back-tracking linesearch see also [40, Algo. 3.1].

As a result, this should yield faster convergence that the simpler LS Algo.1. Trial step length is updated as the minimizer of the quadratic q with $q(0) = \phi(0)$, $q'(0) = \phi'(0)$ and $q(\alpha_i^{(k)}) = \phi(\alpha_i^{(k)})$, c.f. [40, Eqn 3.57]. The initial step length is chosen as follows

$$\beta_{\text{trial ; initial}} \;:=\; \begin{cases} \min\left(\beta_{\text{ref}} \; , \; \beta_{\max}\right) \; . & k = 1 \\ \min\left(\; \alpha_{k-1} \, \dfrac{\phi'_{k-1}(0)}{\phi'_k(0)} \;\; , \;\; \beta_{\text{ref}} \right) & k > 1 \end{cases} \tag{42}$$

where

$$\beta_{\max} \;:=\; -\frac{\phi_k(0)}{\mu \, \phi'_k(0)} \quad ; \quad \beta_{\text{ref}} \;:=\; \frac{\substack{\text{Reference} \\ \text{steplength}}}{\|\mathbf{s}_k\|} \, . \tag{43}$$

Remark 4. *For our tests, $\mu = 0.0001$. In general $\mu < \frac{1}{2}$, c.f. discussion in [39] (so that the criteria is satisfied when the cost function is a quadratic).*

Linesearch Algo 3 is the most expensive of the three algorithms presented here, and requires the **strong Wolfe conditions** which comprise of

$$\begin{aligned} \phi(\alpha) \;&\leq\; \phi(0) \;+\; \mu \; \alpha \; \phi'(0) \quad , \quad \mu \in (0,1) \qquad &&\textbf{Armijo condition} \; ; \\ |\phi'(\alpha)| \;&\leq\; \eta \; |\phi'(0)| \quad , \quad 0 < \mu \leq \eta \qquad &&\textbf{Curvature condition} \; . \end{aligned} \tag{44}$$

The first condition controls the decrease in $\widehat{J}$ and is also called the **sufficient decrease condition**, while the second one prevents the trial step length α from being too small.

The actual algorithm comprises of an outer algorithm described in Algo 3 and an inner one called Zoom Algo 4. We follow the principles of [40, Algo 3.5 and 3.6], and specify how the interval was updated in our implementation. The overall goal of Algo. 3 is to construct (or update to) an interval whose endpoints satisfy the following condition, and to which Zoom Algo 4 is applied.

Condition 1 (on the variables of the zoom algorithm `Zoom`$(\beta_{\text{lo}}, \beta_{\text{hi}})$)**.**

(i) Among all step lengths generated so far that satisfies the sufficient decrease condition, β_{lo} gives the smallest value of the function,

(ii) β_{hi} is chosen so that $\phi'(\beta_{lo})(\beta_{hi} - \beta_{lo}) < 0$.

```
Set $\beta_{\text{trial}} = \beta_{\text{initial trial}}$ with the latter defined in (42);
Set $i = 0$ ;
while $(i < N_{LS\ IterMax\ 1})$ do
    if $\phi(\beta_{trial}) < \phi(0) + \mu\,\beta_{trial}\,\phi'(0)$ then
        $\beta_\star = \beta_{\text{trial}}$ ;
        LS successful ;
        Exit;
    else if $i = N_{LS\ IterMax\ 1} - 1$ then
        $\beta_\star = \beta_{\text{trial}}$ ;
        LS unsuccessful ;
        Exit;
    else
        $\beta_{\text{temp}} = \dfrac{-\phi'(0)\ \beta_{\text{trial}}^2}{2\ [\phi(\beta_{\text{trial}}) - \phi(0) - \phi'(0)\,\beta_{\text{trial}}]}$;
        if $\|\beta_{temp} - \beta_{trial}\| < \epsilon_{Stag\ Steplength}$ then
            $\beta_{\text{trial}} \leftarrow \frac{1}{2}\beta_{\text{trial}}$ ;
        else
            $\beta_{\text{trial}} \leftarrow \beta_{\text{temp}}$ ;
        end
        $i \leftarrow i + 1$;
    end
end
Result: Set $\alpha_k = \beta_\star$
```

Algorithm 2: Enhanced backtracking linesearch with sufficient decrease and second-order interpolation update.

(iii) the interval bounded by β_{lo} and β_{hi} contains step lengths that satisfies the strong Wolfe condition,

Either one enters the Zoom Algo 4, or Algo 3 produces increasing sequence $\beta_1, \beta_2, \ldots$ with properties: $\phi(\beta_i)$ is a decreasing sequence, β_i satisfies the sufficient decrease condition, but fails the curvature condition, see Remark 5 and 6. Once one enters Zoom Algo 4, this means an interval that satisfies Cond. 1 has been constructed. From this step on, either the algorithm stops upon finding the right steplength or hitting the maximum number of iteration $N_{\text{IterMax LS 2}}$, or the Zoom Algo 4 keeps updating to a new interval that also satisfies Cond. 1, see Remark 7.

Remark 5. *In other words, either one enters in the* ***Zoom*** *algorithm, or the above algorithm will produce increasing sequence $\beta_1, \beta_2, \ldots$ with properties: β_i satisfies the sufficient decrease condition, but fails the curvature condition with $\phi'(\beta_i) < 0$, i.e.*

$$\phi(\beta_i) \ < \ \phi(0) + \beta_i\,\mu\,\phi'(0) \quad , \quad \phi'(\beta_i) \ < \ \eta\,\phi'(0)\,.$$

In addition, from comments 1 in Algorithm 3, $\phi(\beta_i)$ is a decreasing sequence. Thus β_k satisfies Condition 1 (i).

Remark 6. *Without $N_{LS\ IterMax\ 2}$ (case 5), the process is repeated (case 3), until an interval with endpoints satisfying the requirements to use the* ***Zoom*** *algorithm (case 1 and 2) is constructed.*

For entering `Zoom` *in case 1, steplength β_{i-1} already satisfies Condition 1 (i). Also from previous remark, $\phi'(\beta_{i-1}) < 0$, and by construction $\beta_{trial} \geq \beta_{i-1}$, we have $\phi'(\beta_{i-1})(\beta_{trial} - \beta_{i-1}) < 0$, thus Condition 1 (ii) is satisfied. It remains to verify Condition 1 (iii). In this case,*

```
Set $\beta_{\text{trial}} = \beta_{\text{initial trial}}$ with the latter defined in (42);
Set $i = 1$ ; Set $\beta_0 = 0$ ;
while $(i < N_{LS\ IterMax\ 1})$ do
    Evaluate $\phi(\beta_{\text{trial}})$.
    if $(\phi(\beta_{trial}) > \phi(0) + \mu\,\beta_{trial}\,\phi'(0))$ or $(\phi(\beta_{trial}) > \phi(\beta_{i-1}))$ then
        $\beta_\star = $ Zoom$(\beta_{i-1}, \beta_{\text{trial}})$ ;                      /* Case 1 */
        exit ;
    else
        /* In this case $\beta_{\text{trial}}$ satisfies s.d.c and $\phi(\beta_{\text{trial}}) < \phi(\beta_{i-1})$ */
        Evaluate $\phi'(\beta_{\text{trial}})$ ;
        if $|\phi'(\beta_{trial})| \leq -\eta\,\phi'(0)$ then
            $\beta_\star = \beta_{\text{trial}}$ ;                                         /* Case 4 */
            Outer_LS = 'successful' ;
            Exit
        else if $\phi'(\beta_{trial}) \geq 0$ then
            $\beta_\star = $ Zoom$(\beta_{\text{trial}}, \beta_{i-1})$ ;    /* Case 2 ( $\Leftrightarrow \phi'(\beta_{\text{trial}}) > -\eta\phi'(0)$ ) */
            Exit
        else if $i = N_{LS\ IterMax\ 1} - 1$ then
            $\beta_\star = \beta_{\text{trial}}$ ;                                         /* Case 5 */
            Outer_LS = 'unsuccessful' ;
            Exit
        else
            $\beta_i = \beta_{\text{trial}}$ ;    /* Case 3 : Interval Update ( $\Leftrightarrow \phi'(\beta_{\text{trial}}) < \eta\phi'(0) < 0$ ) */
            /* The above assignment results in $\beta_i$ satisfying s.d.c and $\phi(\beta_i) < \phi(\beta_{i-1})$ */
            $\beta_{\text{trial}} \leftarrow \min(1.5 \times \beta_{\text{trial}}\ ,\ \beta_{\text{ref}})$ ;           /* Update $\beta_{\text{trial}}$ */
            $i \leftarrow i + 1$ ;
        end
    end
end
Result: Set $\alpha_k = \beta_\star$
```

Algorithm 3: (Outer) Linesearch algorithm using Strong Wolfe conditions to determine step length α_k used to update $\boldsymbol{p}_k$ to $\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \alpha_k \mathbf{s}_k$.

- *Either* $\phi(\beta_{i-1}) < \phi(\beta_{trial})$*, from Remark 5,*

  $$\phi'(\beta_{i-1}) < 0 \quad , \beta_{trial} > \beta_{i-1} \quad \Rightarrow \phi'(\beta_{i-1})\,(\beta_{trial} - \beta_{i-1}) < 0\,.$$

  This means that the conditions of Theorem 6 are satisfied, which implies Condition 1 (iii) is satisfied.

- *Or* β_{trial} *does not satisfy sufficient descent condition (s.d.c). In this case, the endpoints satisfy the assumption of Proposition 7 (with* β_{i-1} *playing the role of* $\gamma_{\min}$ *and* β_{trial} *of* $\gamma_{\max}$*), and thus Condition 1 (iii) is satisfied.*

When entering in `Zoom` *with case 2, both* β_{trial} *and* β_{i-1} *satisfy s.d.c without satisfying the c.c; however we switch the direction of* `Zoom`*, in direction* $\overrightarrow{\beta_{trial}, \beta_{i-1}}$ *(rather than* $\overrightarrow{\beta_{i-1}, \beta_{trial}}$ *as in case 1) since* $\phi(\beta_{trial}) < \phi(\beta_{i-1})$*, thus Condition 1 (i) is satisfied. Condition 1 (iii) is obtained by the asumptions of Theorem 5 are satisfied with*

$$\phi(\beta_{trial}) < \phi(\beta_{i-1}) \quad , \quad \beta_{trial} > \beta_{i-1}, \phi'(\beta_{trial}) \geq 0 \Rightarrow \phi'(\beta_{trial})\,(\beta_{i-1} - \beta_{trial}) < 0\,.$$

As for Condition 1 (ii), since $\phi'(\beta_{trial}) > 0$ *and* $\beta_{trial} > \beta_{i-1}$ *we have* $\phi'(\beta_{trial}(\beta_{i-1} - \beta_{trial}) < 0$.

```
Data: $\phi(0), \phi'(0)$, $\beta_{\rm lo}$, $\beta_{\rm hi}$ satisfying Conditions 1 (i)-(iii)
Result: $\beta_\star$ = Zoom($\beta_{\rm lo}$, $\beta_{\rm hi}$)
Initialization : $\gamma_{\rm lo} \leftarrow \beta_{\rm lo}$ ; $\gamma_{\rm hi} \leftarrow \beta_{\rm hi}$ ; $j \geq 0$;
while $j < N_{LS\ IterMax\ 2}$ do
    if $\|\gamma_{lo} - \gamma_{hi}\| < \epsilon_{Stag\ Steplength}$ then
        $\beta_\star = \gamma_{\rm lo}$ ;
        Zoom_LS = 'unsuccessful' ;
        Exit ;
    else
        $\gamma_{\rm trial}$ = QuadInterpol($\gamma_{\rm lo}, \gamma_{\rm hi}$) defined in 45 ;          /* Update $\gamma_{\rm trial}$ */
        Set $j \leftarrow j + 1$ ;
    end
    if $(\phi(\gamma_{trial}) > \phi(0) + \mu\,\gamma_{trial}\,\phi'(0))$ or $(\phi(\gamma_{trial}) > \phi(\gamma_{lo}))$ then
        /* Interval update case 1 In this case, since $\gamma_{\rm trial}$ is between $\gamma_{\rm lo}$ and $\gamma_{\rm hi}$, we have
           $\phi'(\gamma_{\rm lo}(\gamma_{\rm hi} - \gamma_{\rm lo})\ \phi'(\gamma_{\rm lo}(\gamma_{\rm trial} - \gamma_{\rm lo}) > 0$.  Since $\gamma_{\rm lo}$ and $\gamma_{\rm hi}$ are always chosen so that
           $\phi'(\gamma_{\rm lo})(\gamma_{\rm hi} - \gamma_{\rm lo}) < 0$ (Condition 1 (ii)).  This means that $\phi'(\gamma_{\rm lo}(\gamma_{\rm trial} - \gamma_{\rm lo}) < 0$.  Thus we
           carry out Zoom in the direction $\overrightarrow{\gamma_{\rm lo}, \gamma_{\rm trial}}$.                                  */
        Set $\gamma_{\rm hi} \leftarrow \gamma_{\rm trial}$
    else
        Evaluate $\phi'(\gamma_{\rm trial})$;
        if $|\phi'(\gamma_{trial})| \leq -\eta\,\phi'(0)$ then
            $\beta_\star = \gamma_{\rm trial}$ ;
            Zoom_LS = 'successful' ;
            Exit ;
        end
        /* In this case $\gamma_{\rm trial}$ satisfies s.d.c but fails c.c; that $\phi(\gamma_{\rm trial}) < \phi(\gamma_{\rm lo})$ is automatic since
           $\gamma_{\rm trial}$ is defined by quadratic interpolation.  As a result, $\gamma_{\rm trial}$ can take on the role of $\gamma_{\rm lo}$.
           Since $\gamma_{\rm trial}$ is between $\gamma_{\rm lo}$ and $\gamma_{\rm hi}$, we have $\begin{cases} \phi'(\gamma_{\rm trial})(\gamma_{\rm hi} - \gamma_{\rm lo})\ \phi'(\gamma_{\rm trial})(\gamma_{\rm lo} - \gamma_{\rm trial}) < 0 \\ \phi'(\gamma_{\rm trial})(\gamma_{\rm hi} - \gamma_{\rm lo})\ \phi'(\alpha_{\rm trial})(\gamma_{\rm hi} - \gamma_{\rm trial}) > 0 \end{cases}$.
           The search interval is updated either in direction $\overrightarrow{\gamma_{\rm trial}, \gamma_{\rm lo}}$ (Case 2) or $\overrightarrow{\gamma_{\rm trial}, \gamma_{\rm hi}}$ (Case 3)
           */
        if $\phi'(\gamma_{trial})\,(\gamma_{hi} - \gamma_{lo}) \geq 0$ then
            Set $\gamma_{\rm hi} \leftarrow \gamma_{\rm lo}$ ;   /* Interval update case 2 (in this case $\gamma_{\rm hi}$ is replaced.)   */
        end
        Set $\gamma_{\rm lo} \leftarrow \gamma_{\rm trial}$ ; /* Interval update case 3 (in this case $\gamma_{\rm hi}$ remains the same ) */
    end
    if $j = N_{LS\ IterMax\ 2}$ then
        Zoom_LS = 'unsuccessful' ;
        $\beta_\star = \gamma_{\rm lo}$ ;
    end
end
```

Algorithm 4: Zoom algorithm

Remark 7. *Suppose the previous interval satisfies Condition 1 -3 , each new interval* $\{\alpha_{lo}, \alpha_{hi}\}$ *using the interval update in Zoom Algo 4 satisfies automatically condition 1 and 3. To see condition 2,*

- *Case 1 : we used* $\overrightarrow{\alpha_{lo}, \alpha_{trial}}$ *:* α_k *violates the sufficient decrease condition.*
- *Case 2 and 3 : we used* $\overrightarrow{\alpha_{trial}, \alpha_{lo}}$ *and* $\overrightarrow{\alpha_{trial}, \alpha_{hi}}$ *respectively. By interpolation, we have* $\phi(\alpha_{lo}) > \phi(\alpha_{trial})$ *and* $\phi(\alpha_{hi}) > \phi(\alpha_{trial})$ *.*

Remark 8 (`QuadIterpol`)**.** *The quadratic interpolation* q *with* $q(\alpha_{lo}) = \phi(\alpha_{lo}))$ $q'(\alpha_{lo}) = \phi'(\alpha_{lo}))$ *and* $q(\alpha_{hi}) = \phi(\alpha_{hi}))$ *is given by*

$$q = \left[\frac{\phi(\alpha_{hi}) - \phi(\alpha_{lo}) - (\alpha_{hi} - \alpha_{lo})\phi'(\alpha_{lo})}{(\alpha_{hi} - \alpha_{lo})^2} \right] (\alpha - \alpha_{lo})^2 + \phi'(\alpha_{lo})\,(\alpha - \alpha_{lo}) + \phi(\alpha_{lo})$$

$$q' = 2 \left[\frac{\phi(\alpha_{hi}) - \phi(\alpha_{lo}) - (\alpha_{hi} - \alpha_{lo})\phi'(\alpha_{lo})}{(\alpha_{hi} - \alpha_{lo})^2} \right] (\alpha - \alpha_{lo}) + \phi'(\alpha_{lo})$$

thus the minimizer is given by

$$\alpha_{lo} - \frac{\phi'(\alpha_{lo})\,(\alpha_{hi} - \alpha_{lo})^2}{2\left[\phi(\alpha_{hi}) - \phi(\alpha_{lo}) - (\alpha_{hi} - \alpha_{lo})\phi'(\alpha_{lo})\right]}\,. \tag{45}$$

4.2 Frequency-hopping inversion procedure

Our overall inversion procedure comprises of running gradient-based optimizations at several frequencies, going from low frequencies to high ones. The gradient-based procedure at each frequency is described in Figure 2. At the lowest frequency, the algorithm uses (user-)prescribed initial guess; at successive frequencies, the initial guesses are the reconstructed parameters given by the algorithm at the previous frequency. This strategy is called frequency-hopping and have been used in several areas of inverse problems to correct the ill-posedness of the inverse problem, for reconstruction of conductivity see [8], location of homogeneities [24, 5, 9], in seismic inversion [17]. In our case, using lower frequencies allows the initial guesses, although far from the true models both in terms of position and nature, to converge to the vicinity of the true position; higher frequencies are essential for taking into account the multiple-scattering effect between the obstacles, and are necessary for convergence to positions hidden (in the middle of) in the configurations. For a more mathematical discussion of the frequency-hopping in the context of shape reconstruction, see [44].

We investigate the effectiveness of two search directions: Broyen-Fletcher-Goldfarb-Shannon (BFGS) in the quasi-Newton (QN) family, and Polak-Ribière (PR) with restart in the nonlinear-conjugate gradient (NLCG), in combination with three line search algorithms: Linesearch (LS) Algo. 1 also called simple-backtracking is the cheapest, LS Algo. 2 imposes sufficient descent and second-order quadrature for update, and the most expensive one is strong Wolfe LS Algo. 3. The linesearch algorithms are described in more details in subsection 4.1. For general motivation and history of NLCG, we refer to [25], and of quasi-Newton to [15]. The search directions are chosen with the goal of being able to retrieve as many obstacles as possible while using only first-order information (i.e. the gradient of the cost function), at the same time allowing the initial guesses to be far from the true models. Nonlinear-conjugate gradient and the simpler version, Steepest Descent, are popular in seismic inversion since there is no need for matrix storage, a useful feature for problems which aim to retrieve a large number of parameters. Successful reconstruction for velocity in heterogeneous medium has been shown using NLCG, see results in [17]. Gradient-based optimization, in particular variants in the NLCG family, has been investigated in the context of inverse scattering, see e.g. [45, 31, 17]. A more extensive comparison between the variants (in NLCG) is carried out in [48]. These references suggest the effectiveness of PR. For this reason, we choose PR among others variants in NLCG. Quasi-Newton is less computational intensive than methods in the Newton-like family used in e.g. [36, 33, 35, 31, 8] and the references therein, while still offers good convergence rate of Newton's method [36]. In the references just cited, the linear problem is first linearized as $\Phi(\boldsymbol{p}_{\text{initial guess}}) + \Phi'(\boldsymbol{p}_{\text{initial guess}})\delta_{\boldsymbol{p}} = \mathbf{d}_{\text{obs}}$ and one solves for $\delta_{\boldsymbol{p}}$. This approach is equivalent to employing the Gauss-Newton method in FWI[9].

For the explicit implementation of the search directions, we follow [40, p. 140] for BFGS and and [40, Eqn (5.44), (5.45) p.122-123] for PR with restart (called PR+). The restart, in which the direction is set to Steepest Descent direction, is needed, since the defined PR search direction ([40, Eqn (5.44) p.122]) is not guaranteed to be a descent direction. On the other hand, BFGS implemented with strong Wolfe linesearch algorithm (see below) guarantees descent direction. We use [40, Eqn 6.20] for the initial approximate for the Hessian used in BFGS.

Remark 9. *SD2 has a built in reset to guarantee descent search direction, thus it can be used with all the line search algorithms. This not the case for SD1, and it sometimes can fail with LS1 or LS2.*

These are the stop criteria employed in Algorithm depicted in Figure 2 to skip to the following frequency or to stop the overall procedure:

- those controlling the smallness of $\nabla\widehat{\mathcal{J}}$ and $\widehat{\mathcal{J}}$: $\epsilon_{\text{critical point}}$ and $\epsilon_{\text{err tol}}$;
- those signaling the stagnation of the algorithm at a frequency: $\epsilon_{\text{Stag Steplength}}$ when new steplength is too close to the previous one, $\epsilon_{\text{Stag Pos}}$ when the new configuration is too close to the current one, and $\epsilon_{\text{Stag }\widehat{\mathcal{J}}}$ when the algorithm does not reduce the value of the cost function J (i.e. it is not descending or remains the same compared to e.g. 5 iterations before);
- number of iterations: N_{IterMax} controls the outer iterations (from one reconstruction to the next), $N_{\text{LS IterMax 1}}$ is used within a linesearch algorithm, and $N_{\text{LS IterMax 2}}$ is only used for Linesearch algorithm 3 (in particular the so-called Zoom algorithm);
- the fact whether a linesearch is successful or not,
- the number of frequencies prescribed by user.

5 Common features of the numerical experiments

True models We will work with both periodic (for 6 in Section 6 and 12 obstacles in Section 7) and random true models (for 12 obstacles in Section 8), although periodic configurations, we do not make use of this fact in our inversion. The radius of the obstacles are 0.5. In each configuration, they are either all soft-scattering or hard-scattering.

Synthetic back-scattered data The back-scattered data , inspired from physical experiments in [7], are created by a transreceiver which acts as both a source (by emitting almost planewaves) and receivers, c.f. Figure 4, or 14. For each angle of incidence, synthetic observed data are collected at 128 points, along a line perpendicular to the angle of radiation. The fixed number of data points is applied to all frequencies. The line along which the receivers are placed is also considered as the borders of the domain in which obstacles truly reside. This type of data is called partial or reduced data [24, 37], rather than full data which are collected all around the

---

[9] Gauss-Newton is like Newton's method using line-search strategy however with the Hessian approximated by $\nabla\widehat{\mathcal{J}}^{\star}\nabla\widehat{\mathcal{J}}$, c.f. [40, p.254-258].

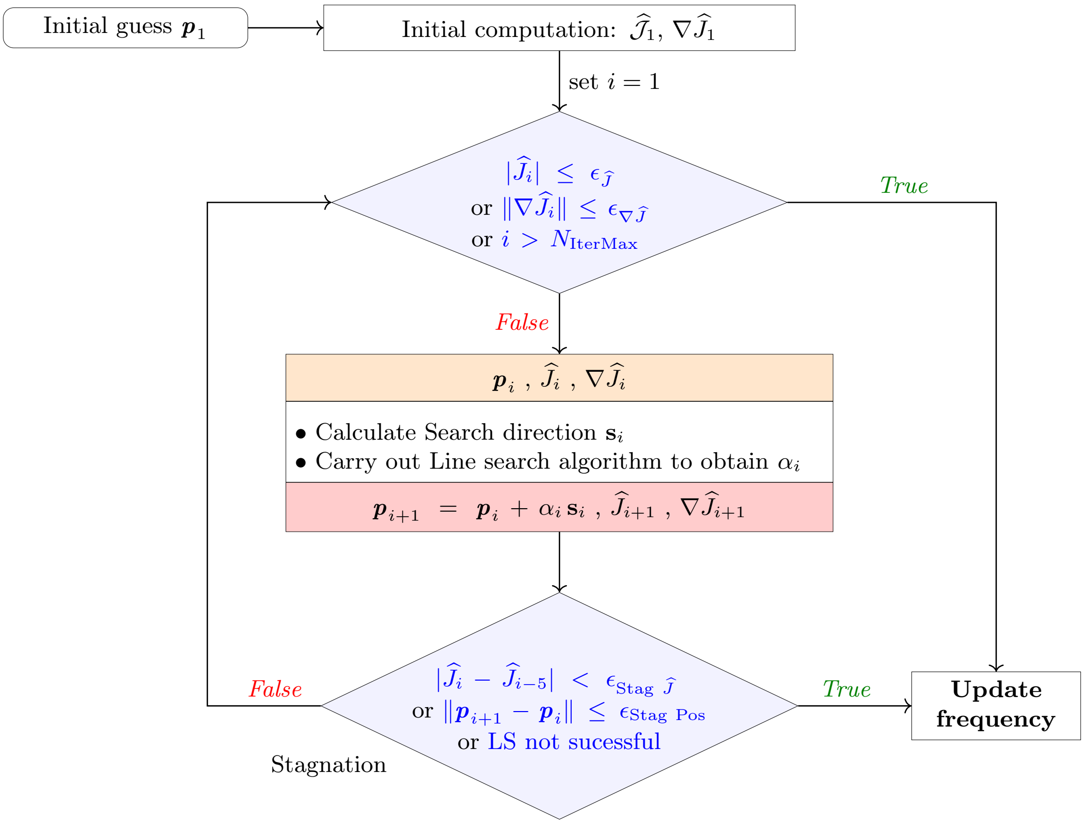


Figure 2: Optimization algorithm at each fixed frequency. At the lowest frequency, the algorithm uses prescribed initial guess; at successive frequencies, the initial guess is the reconstructed parameters given by the algorithm at previous frequency. The values of $\widehat{J}$ and its gradient $\nabla\widehat{J}$ are by the Adjoint-State method, summarized in (21)– (23). Linesearch algorithm uses either Algo. 1, 2 or 3 listed in subsection 4.1.

domain of interest. To create different planewaves (i.e. different angles of incidence), the device is moved to different positions along the edges of the domain of interest.

Backscattered data at multiple frequencies are also used. We use a maximum of 4 angles of radiation (for 12 obstacles). For 6 obstacles with noise-free synthetic data, sucessful reconstruction is obtained with only one angle of radiation, however this means using more frequencies. When using more than three angles of radiation, the range of the frequencies are

$$\kappa_{\text{min}} = 0.08 \leq \;\kappa\; \leq \kappa_{\text{max}} = 0.8 \quad ; \quad 0.04 \;\leq\; \kappa\,\mathsf{r} \;\leq 0.4\,.$$

When only use one angle of radiation (for 6 obstacles), we go up to $\kappa_{\text{max}} = 1.2$ and in this case $\kappa_{\text{max}}\mathsf{r} = 0.6$. These ranges satisfy the requirement for the invertibility, discussed in Footnote 5, of the linear system representing the discretized forward problem.

To avoid the ‘inverse crime’, synthetic data are produced by FSSL order 12 with solver Lapack in a separate set of codes. The solvers used for the resolution of the forward problem is Mumps at lower order of FSSL (≤ 6). For more realistic experiments, complex Gaussian white noise is added by Matlab routine

$$\texttt{awgn}(\text{data}\,,\,\texttt{SNR}_{\texttt{dB}}\,,\,\texttt{`measured'})\,,$$

with $\texttt{SNR}_{\texttt{dB}}$ signal-to-noise ratio per sample in decibel defined as

$$\texttt{SNR}_{\texttt{dB}} = 10 \log_{10} \frac{\|\text{Data Vector}\|}{\|\text{Noise Vector }\|}. \tag{46}$$

Noise Vector is generated using Gaussian probability distribution.

Inversion program The inversion codes are written in `Fortran90`. The experiments are run in sequential on personal laptop. A procedure is designed by indicating in order (in a parameter file to be read by the executive file of the codes) the frequencies to be used, and for each such frequency, the accompanying parameters: order of FSSL (to solve the forward problem), number of angle of radiation, and the accompanying stop and stagnation criteria

$$\epsilon_J \ , \ \epsilon_{\nabla J} \ , \ \epsilon_{\text{Stag J}} \ , \ \epsilon_{\text{Stag Pos}} \ , \ \epsilon_{\text{Stag LS}} \, , \ N_{\text{Iter Max}} \ , \ N_{\text{ LS IterMax 1}}, N_{\text{LS IterMax 2}} \, . \tag{47}$$

The current solver used for the direct problem is Mumps. A more optimal choice will be Lapack, as shown in [7, 6], however for the small linear systems, the performance difference is negligible between these two solvers. Both solvers allow resolution with multiple right-hand-side, which is useful when there are several angles of incidence, i.e. one collects data with several planewaves. The forward linear system (21) becomes

$$\mathbf{A}(\boldsymbol{p}) \ \begin{bmatrix} V^{(1)} & \dots & V^{(N_{\text{Acq}})} \end{bmatrix} \ = \ \begin{bmatrix} F(\boldsymbol{p}, u_{\text{pw}}^{(1)}) & \dots & F(\boldsymbol{p}, u_{\text{pw}}^{(N_{\text{Acq}})}) \end{bmatrix} \ ,$$

and similarly for the adjoint problem (22). Note that both equations use the same LU factorization, another useful feature of Mumps and Lapack already mentioned in Remark 3, and thanks to which the cost of calculating $\widehat{J}$ and its gradient lies mostly in the resolution of the forward problem.

How the inversion procedures are carried out and designed? In initial runs for each configuration and initial guess, the goal is to obtain convergence; as a result, all parameters such as ϵ_J , $\epsilon_{\nabla J}$, $\epsilon_{\text{Stag Pos}}$, $\epsilon_{\text{Stag J}}$ and $\epsilon_{\text{Stag LS}}$ are assigned very small values, and frequencies are added one by one. These parameters are adjusted according if there is still no sign of stagnation of $\widehat{J}$ (by studying the convergence curve). After convergence is obtained, the next step is to refine the results, by increasing the error tolerance and stagnation criteria and by removing redundant high frequencies, in a way so that the same convergence results are obtained. This is done in order to reduce the number of iterations. The actual parameters for each experiment are listed in Appendix E.

Remark 10.

- *Once convergence is obtained (the obstacles find the true positions), more frequencies can be indicated in the parameter files read by the code, however the results remain the same, the program will enter a frequency but leaves immediately without incurring any iteration (due to stagnation).*
- *One aims at a certain error in position, and uses the least number of frequencies that will allow this level of error, i.e. if one stops earlier, the reconstruction result will be lower in quality, and if one continues, the gain in precision is negligible and the algorithm stagnates at redundant frequencies. This is why the convergence results of different methods (different search direction and different linesearch algorithm) are listed with the different number of frequencies used.*
- *Sometimes, fewer frequencies can be used (with stagnation parameters adjusted accordingly), however, this can lead to higher iterations numbers.*

Quality of reconstruction The procedure has no information of the true models (except indirectly through the synthetic observed data). The quality of a reconstruction is calculated after the inversion has been carried out (outside of the inversion codes), and is based on : the number of frequencies used, number of iterations taken, error in J and position. Error in position is calculated as the distance between reconstructed model and the true mode;, in particular

$$\text{Err. Pos }(\boldsymbol{p}_{\text{reconstructed}}) \;:=\; \text{dist}(\boldsymbol{p}_{\text{reconstructed}},\boldsymbol{p}_{\text{model}}) \;=\; \min_{1\leq I,J\leq N} \left\| \begin{pmatrix} x_{I,1} \\ x_{I,2} \end{pmatrix} - \begin{pmatrix} \tilde{x}_{J,1} \\ \tilde{x}_{J,2} \end{pmatrix} \right\| . \tag{48}$$

where

$$\boldsymbol{p}_{\text{reconstructed}} = \left\{ \begin{pmatrix} x_{1,1} \\ x_{1,2} \end{pmatrix} \cdots \begin{pmatrix} x_{N,1} \\ x_{n,2} \end{pmatrix} \right\} \quad , \quad \boldsymbol{p}_{\text{model}} = \left\{ \begin{pmatrix} \tilde{x}_{1,1} \\ \tilde{x}_{1,2} \end{pmatrix} \cdots \begin{pmatrix} \tilde{x}_{N,1} \\ \tilde{x}_{I,2} \end{pmatrix} \right\} .$$

The relative error position is the scaled distance with respect to the size of the domain. Thus on a domain $[a,b]_x \times [c,d]_y$,

$$\text{Rel. Err. Pos }(\boldsymbol{p}_{\text{reconstructed}}) \;:=\; \frac{\text{Err. Pos. }(\boldsymbol{p}_{\text{reconstructed}})}{\max\{|b-a|\;,\;|d-c|\}} . \tag{49}$$

Relative error in J of a reconstruction is calculated at the final frequency of the reconstruction

$$\text{RelErrJ}(\boldsymbol{p}_{\text{reconstructed}}) \;=\; \frac{\sqrt{2\,\widehat{J}(\boldsymbol{p}_{\text{reconstructed}})}}{\|\mathbf{d}_{\text{obs}}\|} \qquad \text{at final reconstruction frequency} . \tag{50}$$

Note that when there is noise, $\mathbf{d}_{\text{obs}}$ is the noisy synthetic data.

Remark 11 (Difficulties in comparing the methods)**.** *Fewer number of iterations does not mean shorter CPU time of the overall reconstruction. The number of iterations indicated does not take into account the iterations used for line search. Simpler line search algorithm can take more iterations, however each iteration is much cheaper than one in more complicated algorithms. We shall see that LS2 is a good compromise between LS1 (the simplest) and LS3 (the most complicated).*

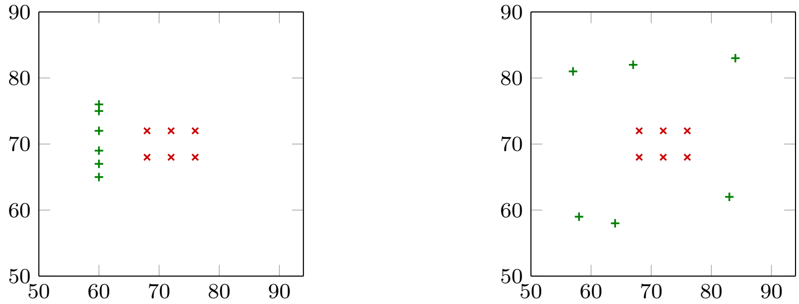
Remark 12. *We have seen that the optimization problem is solved using only first order information, in particular only the value of the cost function and its gradient. However, the value of the cost function and its gradient need to be updated at several points both within each outer iteration and in each linesearch iteration of an outer iteration. This is accomplished via the Adjoint-state method, described in Section 3, each time one needs to solve the forward problem, the computational cost of each update of $\widehat{J}$ and its gradient is hence in the resolution of the corresponding forward problem. In actual experiments, we will see that the number of forward problems is quite large (∼ number of outer iteration × number of linesearch iterations). For experiments with 6 obstacles, they take around 100-150 (outer iterations), this means that the number of actual forward problem that was solved can be up to 30 times this number. This number is higher for experiments with 12 obstacles. Thus the reduction of the resolution time of each forward problem is very important. In our work, this is achieved by using the fast solver FSSL, described in Section 2. We shall see in numerical results that despite the large number of forward problem that need to be solved, the final reconstruction times are in order of 1 sec for 6 obstacles and around 11 secs for 12 obstacles.*

6 Localization of 6 hard-scattering obstacles

In this experiment, we consider the recovery of the position of six hard-scattering obstacles. The domain of interest is of size 42×38 and the (disc-shaped) obstacles have a radius 0.5. As mentioned, we do not assume the periodicity of the true configuration and the centers of the obstacles are at the following location:

$$\begin{array}{ccc} (68,\ 72), & (72,\ 72), & (76,\ 72), \\ (68,\ 68), & (72,\ 68), & (76,\ 68). \end{array}$$

In Figure 3, we illustrate the obstacles configuration for the domain of interest. Figures 3(a) and 3(b) show the two initial guesses considered for the reconstruction. The first one (in Figure 3(a)) is much more different in nature from the true model, than the second one. In Subsection 6.1, we carry out the reconstruction using noise-free data and the initial guess of Figure 3(a). In Subsections 6.2 and 6.3 we incorporate noise in the data (23dB) and test the two initial guesses. All proposed methods are investigated: Quasi-Newton (SD1) and nonlinear conjugate gradient (SD2) search directions with three line search algorithms (backtracking, quadratic interpolation and strong Wolfe condition, respectively LS1, LS2, LS3), for a total of six possibilities.



(a) Initial Guess 1 (IG1): Error Position is 20.5 Relative Error Posisition is 54%

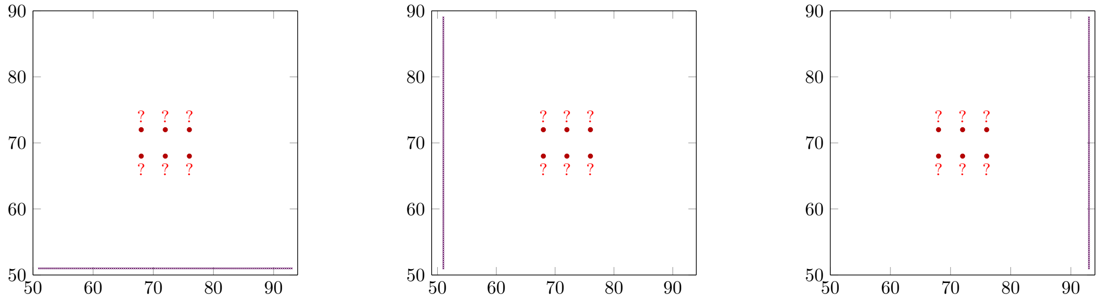
(b) Initial Guess 2 (IG2): Error Position is 29.5, Relative Error Posisition is 77.6 %

Figure 3: Configuration for the six hard-scattering obstacles reconstruction where the obstacles are in red (×) and initial guesses for the reconstruction are in green (+).

In the case with no noise, we only use one angle of radiation for the data: 90°. For the cases where noise is incorporated, three angles are used: 90°, 0° and 180°, see Figure 4. For each angle of incidence, synthetic data are collected at 128 points (*i.e.*, receivers positions) at the boundary, along a line perpendicular to the selected angle of radiation.

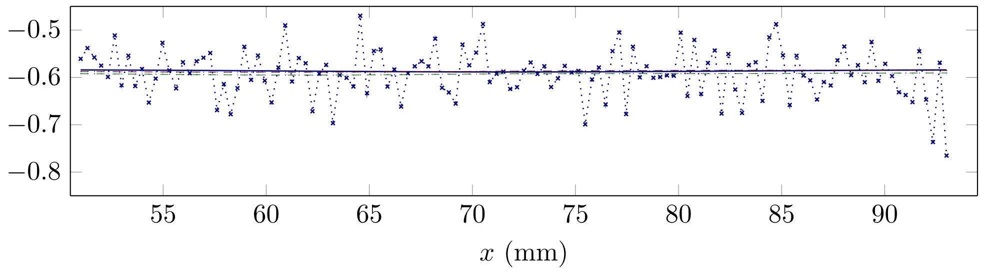
The noisy data with added noise (23dB signal-to-noise ratio, see (46)) correspond to total relative L^2 error (polluted observed data compared to true observed data) between 6.0% and 8.0%, in L^∞ norm between 10% and 21%. For the reconstructions, we take the lowest frequency to be $\kappa = 0.08$, such that $\kappa \mathrm{r} = 0.04$. We illustrate the observed data (with and without noise) as well as the data corresponding to both initial guesses at frequencies $\kappa = 0.08$ and $\kappa = 1.0$ in Figures 5 and 6 respectively.

We see that low frequencies give poor information about the interaction between the obstacles (showing little oscillation) and that, in particular for radiation angle 90°, there is a small gap between the synthetic data (both noise-free and polluted) and the simulated data of the initial

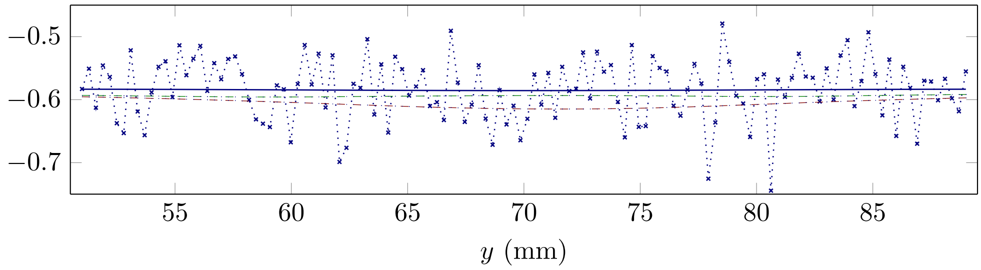


(a) 90° incidence (b) 0° incidence (c) 180° incidence

Figure 4: For each angle of incidence, synthetic data are collected at 128 receivers (·) equally spaced and lying on a line perpendicular to the angle of incidence. These are corresponding sides of domain $[51, 93]_x \times [51, 89]_y$. The goal is to retrieve the positions of the obstacles indicated in red ×.



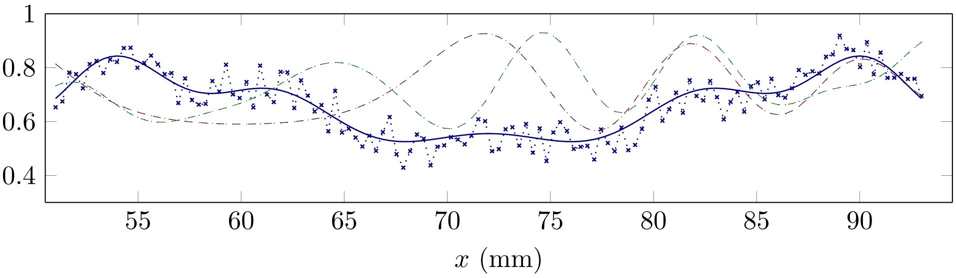
(a) Real part of the data for an incidence angle of 90°. Relative error between observed and noisy data is 7.2% with L^2-norm and 18.2% with L^∞ norm.



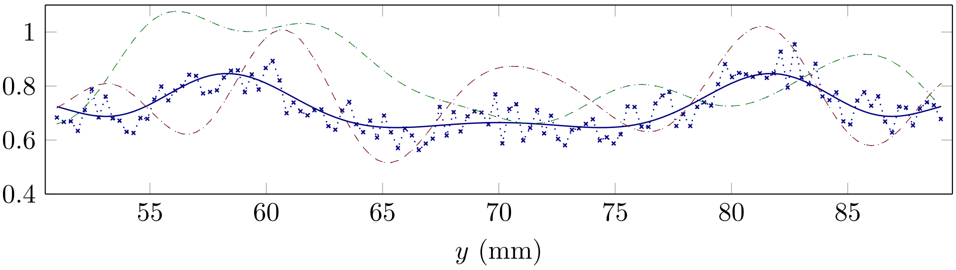
(b) Real part of the data for an incidence angle of 0°. Relative error between observed and noisy data is 6.9% with L^2-norm and 16% with L^∞ norm.

Figure 5: Data obtained at the 128 receivers at frequency $\kappa = 0.08$. The blue line (——) represents the synthetic data, the blue dashed line with crosses (···×···) the data incorporating 23dB noise, the red dashed line (- - -) represents simulated data associated to the initial guess 1 (see Figure 3(a)) and the green dashed line (- - -) represents simulated data associated to initial guess 2 (see Figure 3(b)).

guesses (see Figure 5(a)). This helps explain intuitively how beginning the reconstruction at low frequencies allow that the initial guess to be far from the true model and of different nature (our



(a) Real part of the data for an incidence angle of 90°. Relative error between observed and noisy data is 7.2% with L^2-norm and 18.2% with L^∞ norm.



(b) Real part of the data for an incidence angle of 0°. Relative error between observed and noisy data is 6.9% with L^2-norm and 16% with L^∞ norm.

Figure 6: Data obtained at the 128 receivers at frequency $\kappa = 1$. The blue line (——) represents the synthetic data, the blue dashed line with crosses (···×···) the data incorporating 23dB noise, the red dashed line (- - -) represents simulated data associated to the initial guess 1 (see Figure 3(a)) and the green dashed line (- - -) represents simulated data associated to initial guess 2 (see Figure 3(b)).

initial guess 2 is not at all structured). At higher frequency, the data collected for the different configurations provide a much larger contrast (see Figure 6).

6.1 Inversion using noise-free data

We first work with noise-free data, starting from IG2 (Figure 3(b)) and using only one angle of illumination: 90°[10] For the reconstruction, κ ranges from 0.08 to 1.2. We test the different methods that have been introduced, namely, two search directions for which three line search algorithms can be used. The information for each case is given Table 1.

We proceed with the iterative procedure, the information on the processes are given in Table 2 and the progression of the scaled position error with iteration is shown in Figure 7. In the Figure 8 we visualize the positions of the obstacles recovered for three selected iterations, using the SD2-LS2 method.

The reconstruction provides a scaled position error of less than 1% in all cases and the reconstructed configuration is almost indistinguishable from the true model (see Figure 8, where the case giving the most error, SD2-LS2, is shown). In this case, where noise-free data are used, all methods converge similarly, in a maximum of 100 iterations. However, we note that the

---

[10] If we use more data associated with other angles of illuminations, then the number of frequencies needed for the reconstruction will reduce.

Search Direction (SD)	Line search (LS)	LS coeff.	$\sharp$ frequency	list of frequency
1	1	n/a	12	0.08 , 0.09 , 0.1 : 0.1: 0.5 , 0.8 : 0.1 : 1.2
1	2	10^{-4},n/a	12	0.08 , 0.09 , 0.1:0.1:1.0
1	3	10^{-4},0.9	12	
2	1	n/a	9	0.08 , 0.09 , 0.1:0.1:0.5 , 0.8 , 0.9
2	2	10^{-4},n/a	12	0.08 , 0.09 , 0.1:0.1:1.0
2	3	10^{-4},0.9	12	

Table 1: Parameter used for the noise-free data experiment for the 6 Obstacles reconstruction using data from one angle of radiation (90°). For each frequency, the order of FSSL is 3. The colon notation x:y:z indicates and interval from x to z with step y. Experiment **[6Obs.NoiseFree.IG2]**.

Method	$\sharp$ freq	$\kappa_{\max}$	Final J	Final Rel. Error J	$\sharp$ iterations	Final position error	Final scaled position error	Run time (s)
SD1-LS1	12	1.2	3.0×10^{-4}	0.21%	77	0.21	0.54%	0.54
SD1-LS2	12	1.0	4.0×10^{-4}	0.25%	72	0.18	0.46%	0.22
SD1-LS3	12	1.0	1.0×10^{-4}	0.13%	89	0.21	0.55%	0.26
SD2-LS1	9	0.9	3.8×10^{-5}	0.08%	82	0.18	0.40%	0.58
SD2-LS2	12	1.0	7.0×10^{-5}	0.11%	100	0.32	0.85%	0.29
SD2-LS3	12	1.0	4.0×10^{-5}	0.08%	84	0.17	0.44%	0.41

Table 2: Information on the reconstructions of the 6 obstacles position using data without noise and different methods. Experiment **[6Obs.NoiseFree.IG2]**

SD2-LS1 method is able to converge using nine frequencies, while it has not been possible with the other methods. Regarding Remark 11, LS2 (used with both search directions) might take more iterations, but the final reconstruction takes the least time. Although LS1 does not take more iteration than LS2 or LS3, the overall reconstruction time is longer because more time is spent on the inner line search iterations.

6.2 Inversion using data with noise – Initial Guess 1

We now incorporate noise in the data with a signal-to-noise ratio of 23dB. The data used cover three angles of radiation: 90°, 0° and 180°, according to Figure 4. The effect of noise on the data is shown in Figures 5 and 6. For the reconstruction, we start from the initial guess 1 (see Figure 3(a)); a set of nine frequencies is used: 0.08, 0.09 and from 0.1 to 0.7 with step 0.1. All methods use the complete set of frequencies, at the exception of the SD2-LS1 (Search Direction method 2, Line Search method 1) where only eight are used, omitting the highest frequency. In Table 3, we provide the information for the iterative reconstruction. The Figure 9 shows the evolution of the error in the position of the reconstructed obstacles with iterations. The Figure 10 shows the evolution of the cost function with frequency iterations for five selected frequencies.

We observe that optimization at lower frequencies stagnates faster than at higher frequencies, see Figure 10. The same figure also shows that the methods SD2-LS1, SD2-LS3 and SD1-LS3

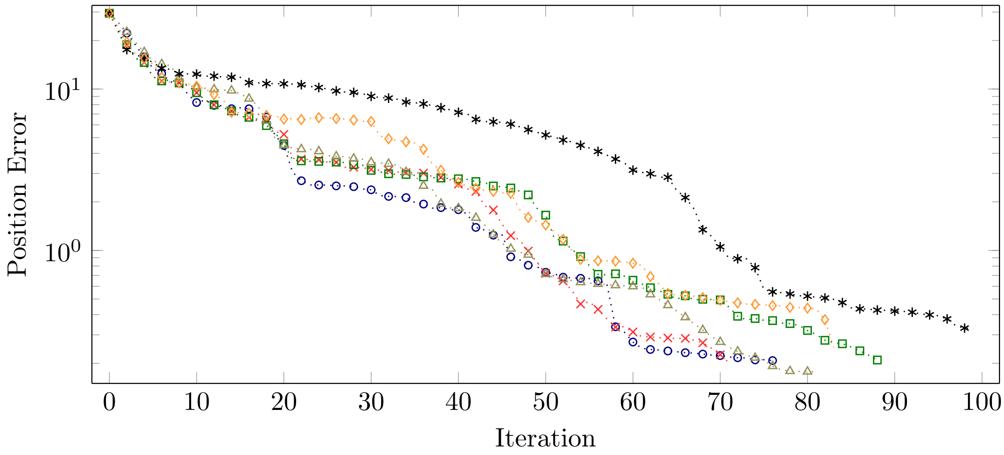
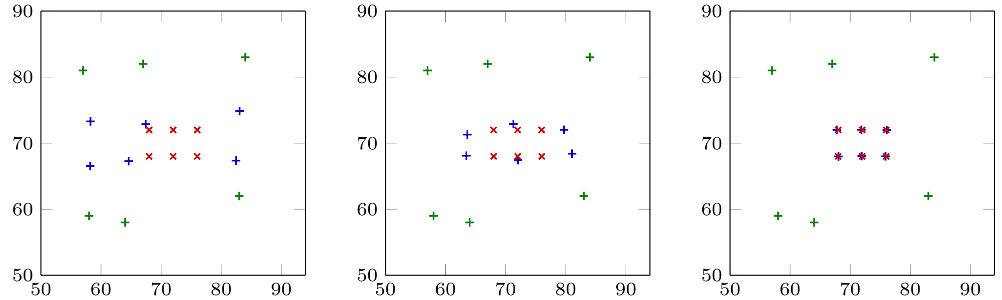


Figure 7: Progression of the obstacles position error with iterations for the reconstruction of 6 obstacles using noise-free data. The markers are only placed one iteration over two for clarity. The blue circles (···○···) indicate the SD1-LS1 method; The red crosses (···×···) indicate the SD1-LS2 method ; The green squares (···□···) indicate the SD1-LS3 method ; The yellow triangles (···△···) indicate the SD2-LS1 method ; The black stars (···∗···) indicate the SD2-LS2 method ; The orange diamonds (···◇···) indicate the SD2-LS3 method. The details of the method parameters are given Tables 1 and 2. Experiment **[6Obs.NoiseFree.IG2]**.



(a) Iteration 2, $\kappa = 0.08$; $J = 6.0 \times 10^{-4}$; position error is 17.56; scaled position error is 46%.

(b) Iteration 30, $\kappa = 0.6$; $J = 0.16$; position error is 2.7; scaled position error is 24%.

(c) Iteration 100, $\kappa = 1$; $J = 8.0 \times 10^{-4}$; position error is 0.3; scaled position error is 0.8%.

Figure 8: Obstacles position recovery using noise-free data with the SD2-LS2 method. Experiment **[6Obs.NoiseFree.IG2]**.

converge faster and require fewer iterations per frequency to obtain a marked decrease in the cost function, compared to the rest of the methods. What is interesting is SD2 with the simplest linesearch LS1 is most efficient, using only 42 global iterations with the lowest time cost of 0.31 secs, see Figure 11 for the illustration of its reconstruction result. The lesser efficient methods for this configuration are SD1-LS1 and SD2-LS2 exhibiting a slower convergence rate and requiring much more iterations, see Figure 9. In any case, two search directions perform similarly well. Although they might take different number of iterations to convergence, all methods are able to arrive at a reconstruction of similar accuracy, as illustrated in Table 3.

SD	LS	LS coeff.	♯ frequency	$\kappa_{\max}$	♯ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	1	n/a	9	0.7	88	0.79	7.00%	0.7%	0.64
1	2	(0.0001, n/a)	8	0.6	67	0.94	6.77%	1.0%	0.4
1	3	(0.0001, 0.9)	9	0.7	72	0.79	7.00%	0.8%	0.6
2	1	n/a	9	0.7	42	0.79	7.00%	1.0%	0.31
2	2	(0.0001, n/a)	9	0.7	131	0.79	7.00%	0.6%	0.72
2	3	(0.0001, 0.9)	9	0.7	59	0.79	7.00%	0.8%	0.82

Table 3: Parameter used for the 6 Obstacles reconstruction using noisy data from three angles of radiation and the initial guess 1. The level of noise in the data is of 23dB. For each frequency, the order of FSSL is 3. SD indicates the search direction method and LS line search. Experiment **[6Obs.23dB.IG1]**.

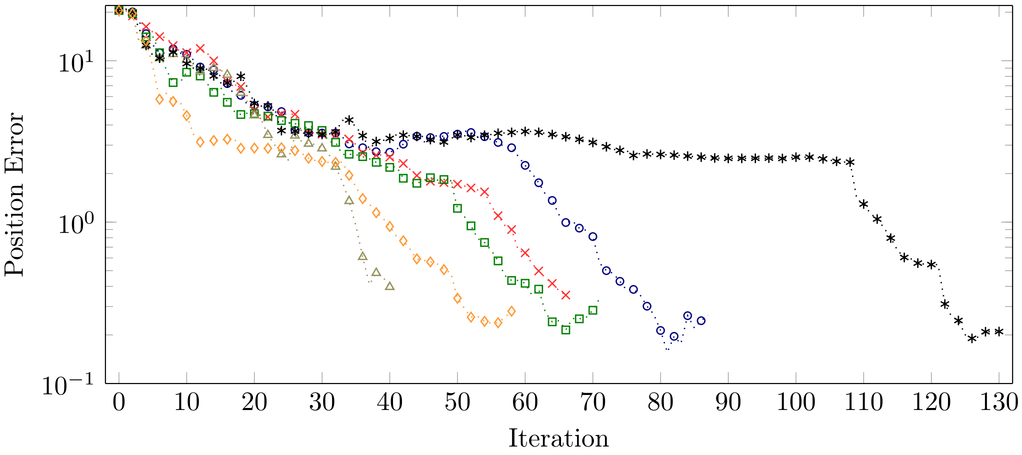


Figure 9: Progression of the obstacles position error with iterations for the reconstruction of 6 obstacles using level of noise of 23dB in the data and starting from the initial guess 1 (Figure 3(a)). The markers are only placed one iteration over two for clarity. The blue circles (···○··) indicate the SD1-LS1 method; The red crosses (···×··) indicate the SD1-LS2 method ; The green squares (···□··) indicate the SD1-LS3 method ; The yellow triangles (···△··) indicate the SD2-LS1 method ; The black stars (···∗··) indicate the SD2-LS2 method ; The orange diamonds (···◇··) indicate the SD2-LS3 method. The details of the method parameters are given Tables 3. Experiment **[6Obs.23dB.IG1]**.

6.3 Inversion using data with noise – Initial Guess 2

We reproduce the previous experiment of recovering six obstacles from noisy data, but starting from the initial guess 2 (see Figure 3(b)). Here we investigate how the starting information influences the reconstruction, in terms of accuracy and convergence speed, depending on the methods. In Table 4, we give the information regarding the different setups and resulting number of iterations and accuracy of the recovery. In Figure 12, we plot the evolution of the position error of the obstacles with iterations. We note that the position error of this initial guess is slightly larger than for the first initial guess, thus we expect the need for more iterations.

The efficiency of the lineasearch algorithms, when coupled with SD2, changes compared to

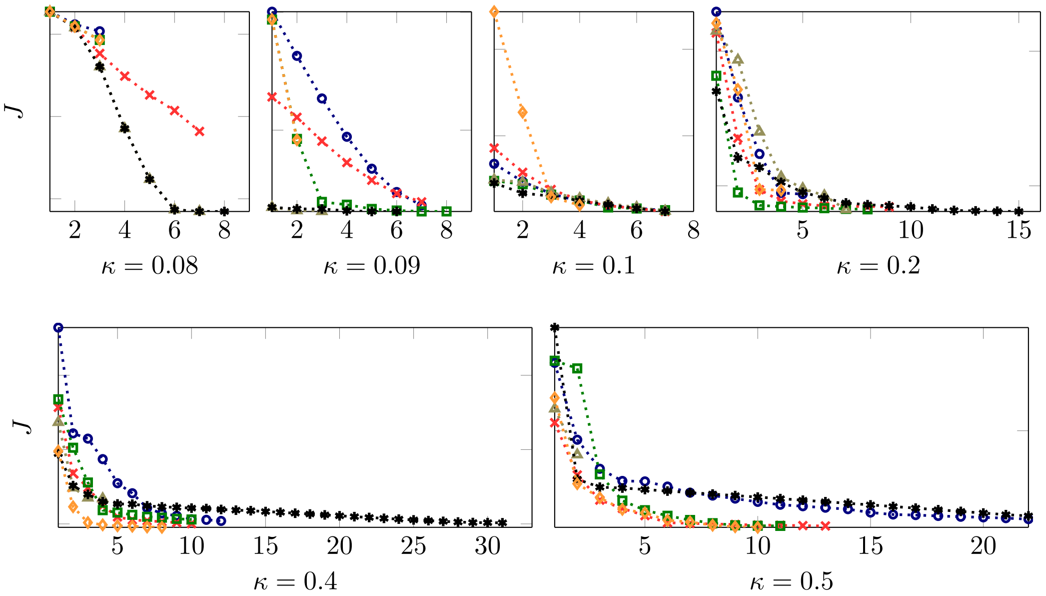
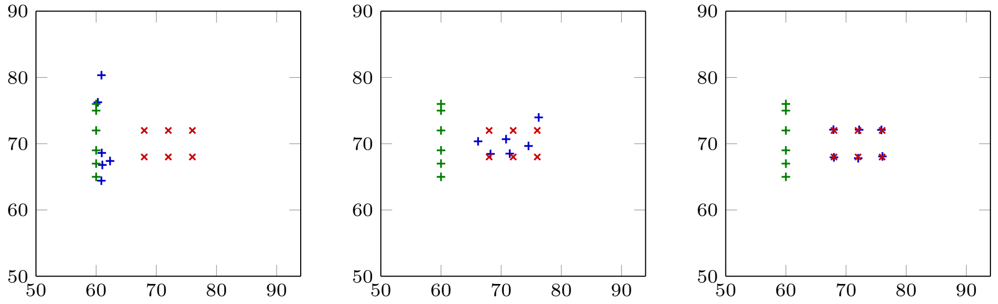


Figure 10: Evolution of the cost function depending on the methods for different frequencies for the reconstruction of 6 obstacles using level of noise of 23dB in the data and starting from the initial guess 1 (Figure 3(a)). The blue circles (···○··) indicate the SD1-LS1 method; The red crosses (···×··) indicate the SD1-LS2 method ; The green squares (···□··) indicate the SD1-LS3 method ; The yellow triangles (···△··) indicate the SD2-LS1 method ; The black stars (···∗··) indicate the SD2-LS2 method ; The orange diamonds (···◇··) indicate the SD2-LS3 method. The details of the method parameters are given Tables 3. All methods do not have the same number of iterations per frequency due to the stagnation check in the iterative algorithm. Experiment **[6Obs.23dB.IG1]**.



(a) Iteration 2, κ =; J =; position error is ; scaled position error is %.

(b) Iteration 21, κ =; J =; position error is ; scaled position error is %.

(c) Iteration 42, κ =; J =; position error is ; scaled position error is %.

Figure 11: Obstacles position recovery using 23dB data with the SD2-LS1 method and starting from initial guess 1 (Figure 3(a)). Experiment **[6Obs.23dB.IG1]**.

the reconstruction starting from the initial guess 1 (see Table 3). While SD2-LS1 using initial guess 1 performs the fastest, with initial guess 2, the convergence of SD2-LS1 and SD2-LS3

SD	LS	LS coeff.	$\sharp$ frequency	$\kappa_{\max}$	$\sharp$ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	1	n/a	9	0.7	72	0.79	7.00%	0.8%	0.67
1	2	(0.0001, n/a)	8	0.6	79	0.94	6.77%	1.0%	0.48
1	3	(0.0001, 0.9)	8	0.6	75	0.94	6.77%	0.8%	0.68
2	1	n/a	10	0.8	181	1.11	6.88%	0.7%	2.64
2	2	(0.0001, n/a)	9	0.7	79	0.79	7.00%	0.75%	0.46
2	3	(0.0001, 0.9)	8	0.6	155	0.94	6.77%	0.73%	1.40

Table 4: Parameter used for the 6 Obstacles reconstruction using noisy data from three angles of radiation and the initial guess 2. The level of noise in the data is of 23dB. For each frequency, the order of FSSL is 3. SD indicates the search direction method and LS line search. Experiment **[6Obs.23dB.IG2]**.

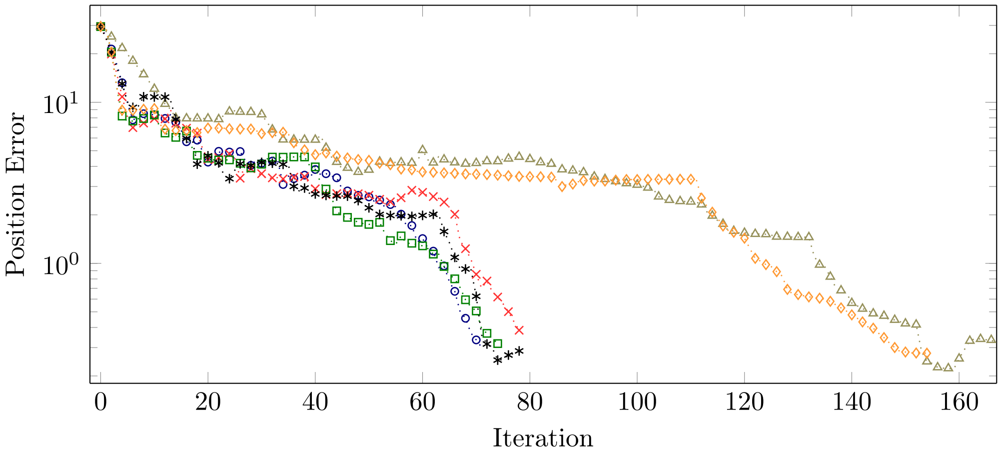
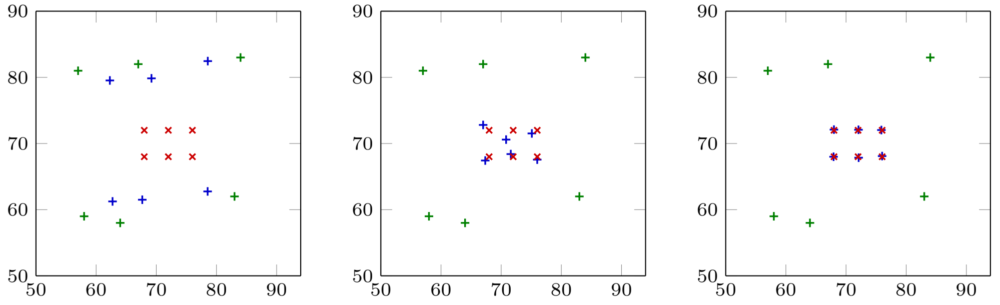


Figure 12: Progression of the obstacles position error with iterations for the reconstruction of 6 obstacles using level of noise of 23dB in the data and starting from the initial guess 2 (Figure 3(b)). The markers are only placed one iteration over two for clarity. The blue circles (···○···) indicate the SD1-LS1 method; The red crosses (···×···) indicate the SD1-LS2 method ; The green squares (···□···) indicate the SD1-LS3 method ; The yellow triangles (···△···) indicate the SD2-LS1 method ; The black stars (···∗···) indicate the SD2-LS2 method ; The orange diamonds (···◇···) indicate the SD2-LS3 method. The details of the method parameters are given Tables 3. Experiment **[6Obs.23dB.IG2]**.

require twice the number of iterations than the SD2-LS2. On the other hand, the behavior of the linesearches coupled with SD1 remain the same for both initial guesses; in particular, the comparison in convergence pattern among SD1-LS1, SD1-LS2 and SD1-LS3 remain the same for both initial guesses. This means that SD1 appears to be better solution, less affected by starting model. Overall, as in with IG1, all methods, while taking different number of iterations, are able to provide similar accuracy in the reconstruction, with less than one percent position error. In Figure 13, we show the position of the obstacles for three iterations using the SD2-LS2 method, where we can see the accuracy of the final reconstruction.



(a) Iteration 2, κ =; J =; position error is ; scaled position error is %.

(b) Iteration 39, κ =; J =; position error is ; scaled position error is %.

(c) Iteration 79, κ =; J =; position error is ; scaled position error is %.

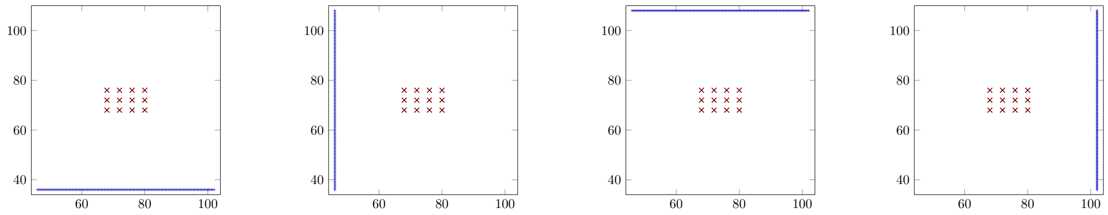
Figure 13: Obstacles position recovery using 23dB data with the SD2-LS2 method and starting from initial guess 2 (Figure 3(b)). Experiment **[6Obs.23dB.IG2]**.

7 Localization of 12 soft-scattering obstacles

In this section, we consider the case of twelve soft-scattering obstacles of radius 0.5 to be recovered. They are positioned in a structured organization with center locations:

$$\begin{array}{cccc} (76,\ 68), & (76,\ 72), & (76,\ 76), & (76,\ 80), \\ (72,\ 68), & (72,\ 72), & (72,\ 76), & (72,\ 80), \\ (68,\ 68), & (68,\ 72), & (68,\ 76), & (68,\ 80). \end{array}$$

Compared to the previous experiments where only six obstacles are considered, we now take four angle of radiation for the data, *i.e.* we generate illumination from all sides of our domain, see Figure 14. In Figure 15, we show the two initial guesses taken to start the reconstruction algorithm.



(a) 90° incidence

(b) 0° incidence

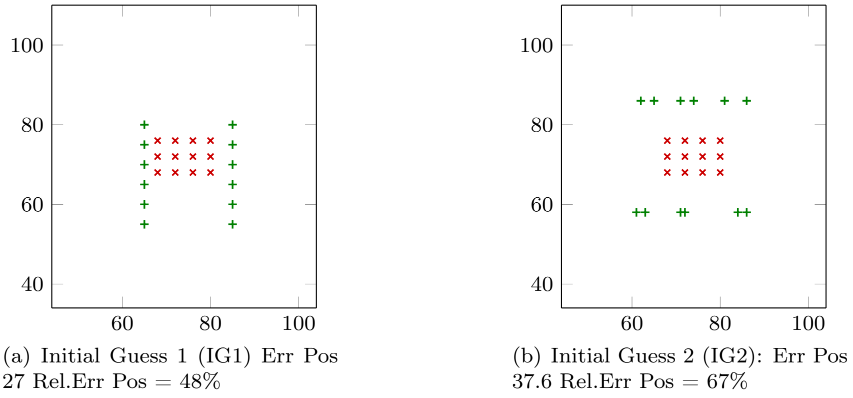
(c) 270° incidence

(d) 180° incidence

Figure 14: For each angle of incidence, synthetic data are collected at 128 receivers (+) equally spaced and lying on a line perpendicular to the angle of incidence. These are corresponding sides of domain $[46,102]_x \times [36,108]_y$. The goal is to retrieve the positions of the twelve obstacles indicated in red (×).

7.1 Inversion using noise-free data

We apply the iterative reconstruction algorithm with noise-free data, and starting from the initial guess 2 (Figure 15(b)). We compare the efficiency of the two search directions by performing



(a) Initial Guess 1 (IG1) Err Pos 27 Rel.Err Pos = 48%

(b) Initial Guess 2 (IG2): Err Pos 37.6 Rel.Err Pos = 67%

Figure 15: Configuration for the twelve obstacles reconstruction where the obstacles are indicated in red (×) and the initial guesses for the reconstruction are in green (+).

the methods SD1-LS3 and SD2-LS1. The algorithms use frequency from $\kappa = 0.09$ to $\kappa = 0.7$ for the SD1-LS3 and up to $\kappa = 0.9$ for the SD2-LS1. In Table 5, we list the information for the two cases.

SD	LS	♯ frequency	$\kappa_{\max}$	♯ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	3	9	0.7	270	6.0×10^{-4}	0.05%	0.1%	7
2	1	9	0.7	300	9.0×10^{-2}	1.90%	2%	23

Table 5: Parameter used for the 12 Obstacles reconstruction using noise-free data starting from initial guess 2 (Figure 15(b)). The order of FSSL method for the modeling is of 3 for frequency lower than 0.4 and order 4 for the higher frequency. SD indicates the search direction method and LS line search. SD2-LS3 line search parameters are $\mu = 10^{-4}$ and $\eta = 0.4$. Experiment **[12Obs.NoiseFree.IG2]**.

We plot the evolution of the cost function with iterations at selected frequencies for both methods in Figure 16. In this figure, at initial frequencies ($0.1 \leq \kappa \leq 0.4$), method SD2-LS1 provides better convergence, as the cost function decreases faster than it does for SD1-LS3. At higher frequencies, although the progression of the cost function has the same pattern for both methods at $\kappa = 0.5$, SD1-LS3 performs much better for higher frequency $\kappa = 0.6$. This difference in performance can also be seen in the evolution of the error in obstacles position after about iteration 150, shown in Figure 17. Also, in this figure, for SD1-LS3, the error position decreases at first slowly (at initial iterations i.e. lower frequencies) before a rapid descent takes place to a final position of 0.1 %. For SD2, this decrease of the error position is much less drastic, and ends at a much higher position error 2%. The quality of the reconstruction are illustrated in Figures 18 and 19, in which distinctly higher precision (20 times higher) is given by SD1-LS3. Furthermore, while a higher quality reconstruction, the time cost of SD1-LS3 is merely a third of SD2-LS1. As a result, between the two methods, SD1-LS3 performs better. This results also implies that although simpler linesearch LS1 can perform well in simpler configurations (see Experiment [6Obs.23dB.IG1] in subsection 6.2), complicated configurations will require more sophisticated linesearchs.

Remark 13. *We have seen from Table 5 that constraint to using the same number of frequencies, method SD2-LS1 provides a much worse result than SD1-LS3, with a precision 20 times less while taking 3 times longer. This method however can attain similar accuracy, given higher frequencies and incurring further computational cost. In fact, it needs 2 more frequencies (0.8 and 0.9) and the final run time is slight higher for 30 secs.*

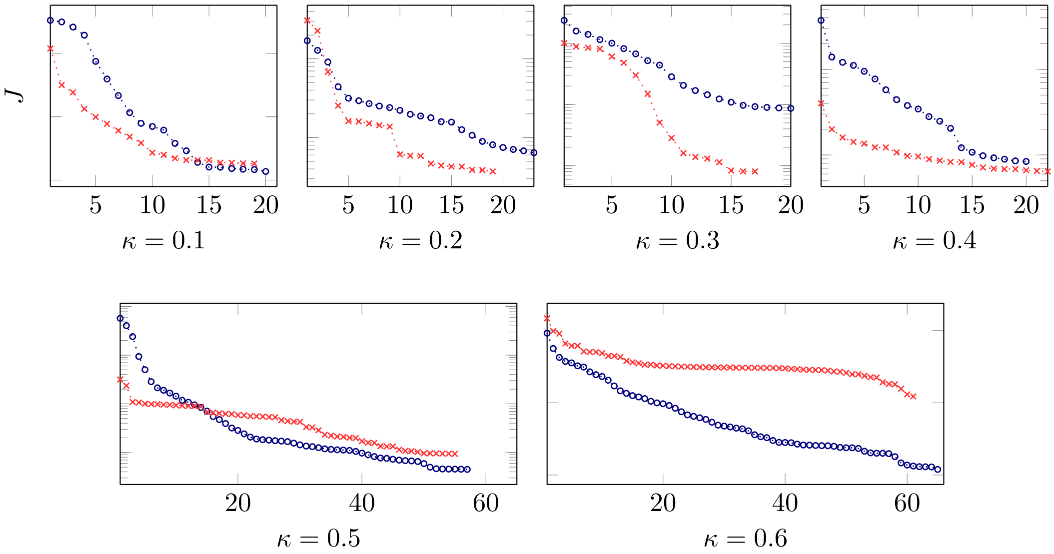


Figure 16: Evolution of the cost function depending on the methods for different frequencies for the reconstruction of 12 obstacles starting from the initial guess 2 (Figure 15(b)). The blue circles (···○···) indicate the SD1-LS3 method; The red crosses (···×···) indicate the SD2-LS1 method ; The two methods do not have the same number of iterations per frequency due to the stagnation check in the iterative algorithm. Experiment **[12Obs.NoiseFree.IG2]**.

7.2 Inversion using data with noise – Initial Guess 1

We now incorporate noise in the generated data, using a signal-to-noise ratio of 30dB, following (46). It leads to a relative L^2-norm error between 2.8% and 3.5% and L^∞-norm between 4.1% and 8.1% compared to the synthetic, noise-free data. We start from the initial guess 1 of Figure 15(a).

We compare two methods for the reconstruction: quasi-Newton Search Direction using strong Wolfe condition for line search (SD1-LS3) and Nonlinear conjugate gradient Search Direction, also using strong Wolfe condition for line search (SD2-LS3). Both methods use the same line search parameters with $\mu = 1.0 \times 10^{-4}$ and $\eta = 0.4$. The frequency varies from $\kappa = 0.09$ to 0.7. Table 6 gives the information regarding the reconstruction runs.

Figures 20 and 21 plot the evolution of the cost function and the position error respectively, and compare the two methods. In Figure 22, we illustrate the obstacles reconstruction with iterations for the SD1-LS3 method. From Figures 20, we observe similar behavior as in the noise-free experiments (Experiment [12Obs.NoiseFree.IG2] in subsection 7.1): at initial frequencies ($\kappa = 0.2, 0.3$), the method SD2-LS3 behaves slightly better than SD1-LS3, with steep drops in the value of the cost function occurring from the beginning of the iterations; at higher frequencies, the progression of the cost function is similar for the two search directions. For most frequencies (except for the last one $\kappa = 0.7$), both methods are very comparable, as indicated by the evolution

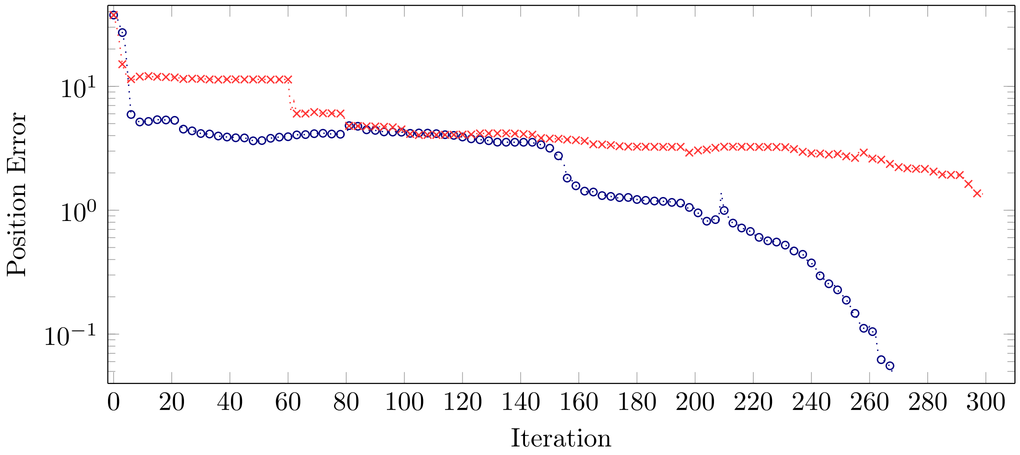
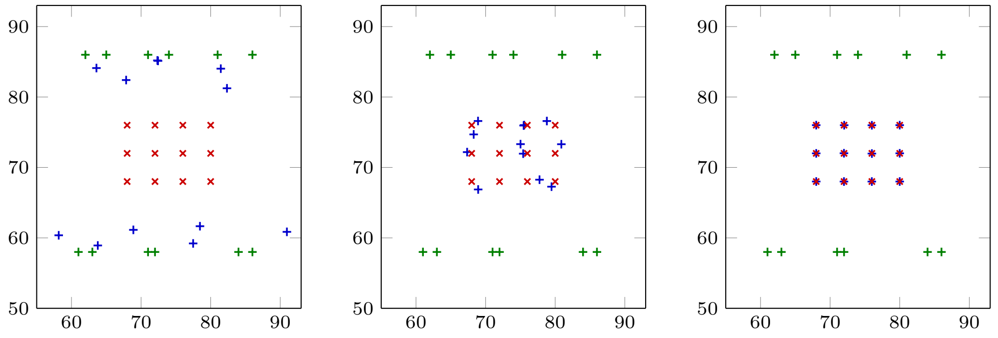


Figure 17: Progression of the obstacles position error with iterations for the reconstruction of 12 obstacles starting from the initial guess 2 (Figure 17). The markers are only placed one iteration over three for clarity. The blue circles (···○··) indicate the SD1-LS3 method; The red crosses (···×··) indicate the SD2-LS1 method. The details of the method parameters are given Table 5. Experiment **[12Obs.NoiseFree.IG2]**.



(a) Iteration 2, $\kappa = 0.1$; $J = 98$; position error is 32 ; scaled position error is 56%.

(b) Iteration 30, $\kappa = 0.2$; $J = 0.02$; position error is 4 ; scaled position error is 7%.

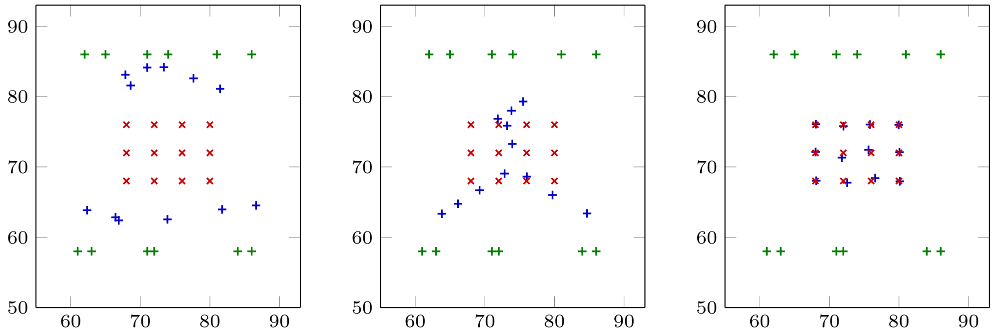
(c) Iteration 270, $\kappa = 0.7$; $J = 6.0 \times 10^{-5}$; position error is 0.04 ; scaled position error is 0.08%.

Figure 18: Twelve obstacles position recovery using noise-free data with the SD1-LS3 method and starting from initial guess 2 (Figure 15(b)). Experiment **[12Obs.NoiseFree.IG2]**.

of position error of Figure 21. At the highest frequency, SD1-LS3 performs slightly better than SD2-LS3, and results in a reconstruction with half position error , see Figure 21. Since the total number of iterations and time cost are similar, with reconstruction of almost comparable precision, see Table 6), the two search directions are comparable, with a slight edge towards SD1.

7.3 Inversion using data with noise – Initial Guess 2

We repeat the experiment using the noisy data but starting from the initial guess 2 of Figure 15(b), which are further away from the actual obstacles, compared to the initial guess 1. We proceed with two methods: quasi-Newton or Nonlinear conjugate gradient, both using strong



(a) Iteration 2, $\kappa = 0.09$; $J = 34$; position error is 23; scaled position error is 40%.

(b) Iteration 30, $\kappa = 0.09$; $J = 13$; position error is 11 ; scaled position error is 21%.

(c) Iteration 300, $\kappa = 0.7$; $J = 9.0 \times 10^{-2}$; position error is 1.35 ; scaled position error is 2%.

Figure 19: Twelve obstacles position recovery using noise-free data with the SD1-LS3 method and starting from initial guess 2 (Figure 15(b)). Experiment **[12Obs.NoiseFree.IG2]**.

SD	LS	$\sharp$ frequency	$\kappa_{\max}$	$\sharp$ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	3	8	0.7	525	0.25	3.18%	0.3%	15
2	3	8	0.7	503	0.26	3.24%	0.7%	13

Table 6: Parameter used for the 12 Obstacles reconstruction using noisy data starting from initial guess 1 (Figure 15(a)). The order of FSSL method for the modeling is of 3 for frequency lower than 0.4 and order 4 for the higher frequency. SD indicates the search direction method and LS line search. Line search parameters are $\mu = 10^{-4}$ and $\eta = 0.4$ for both methods. Experiment **[12Obs.30dB.IG1]**.

Wolfe condition for line search (SD1-LS3 and SD2-LS3 respectively). The line search parameters are $\mu = 1.0 \times 10^{-4}$ and $\eta = 0.4$. As in the previous experiment, the frequency varies from $\kappa = 0.09$ to 0.7. The information for the iterative algorithm are given in Table 7.

The progression of the cost function with iterations is plotted in Figure 23. We observe very similar behavior for the two methods and, as in our previous tests (Experiment [12Obs.NoiseFree.IG2] in subsection 7.1 and Experiment [12Obs.30dB.IG1] in subsection 7.2), the convergence for low frequencies is more rapid with the SD2-LS3 method, while at higher frequency, it is faster with SD1. The two methods are yet very close, as it is indicated from the progression of the position error with iterations in Figure 24. In this figure, one observes that the slow convergence of SD1-LS3 at low frequencies is compensated by the rapid convergence at higher frequencies. With initial guess 2, the difference in performance is imperceptible compared to that for initual guess 2; accuracy, number of iterations, and run time are comparable between the two methods (see Table 7). In the Figure 25, we show the obstacle position recovery at three iterations, for the SD1-LS3 method.

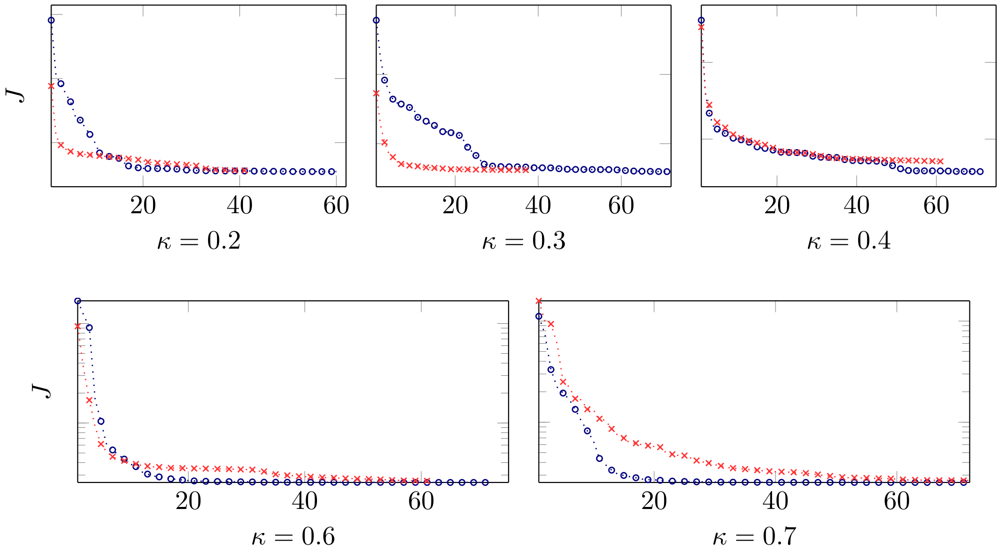


Figure 20: Evolution of the cost function depending for different frequencies for the reconstruction of 12 obstacles using noisy data and starting from the initial guess 1 (Figure 15(a)). The markers are indicated one over two iterations for clarity. The blue circles (···o··) indicate the SD1-LS3 method; The red crosses (···×··) indicate the SD2-LS3 method ; The two methods do not have the same number of iterations per frequency due to the stagnation check in the iterative algorithm. Experiment **[12Obs.30dB.IG1]**.

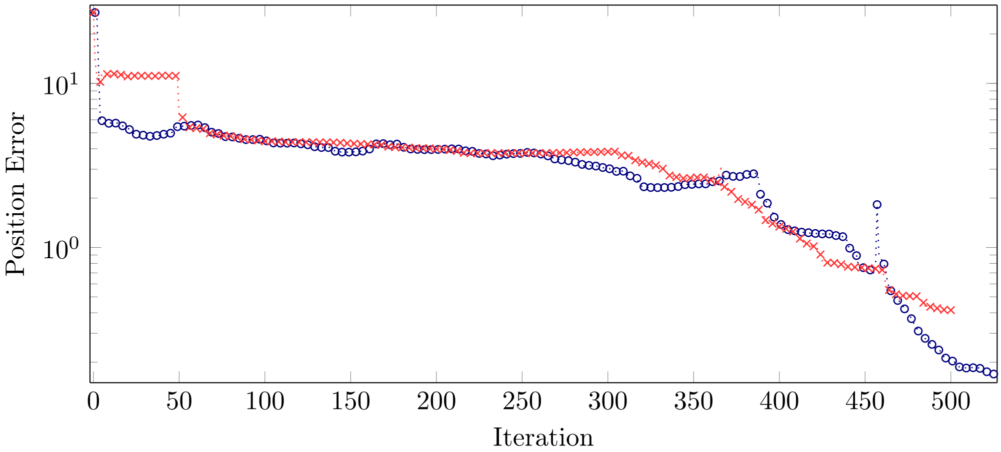
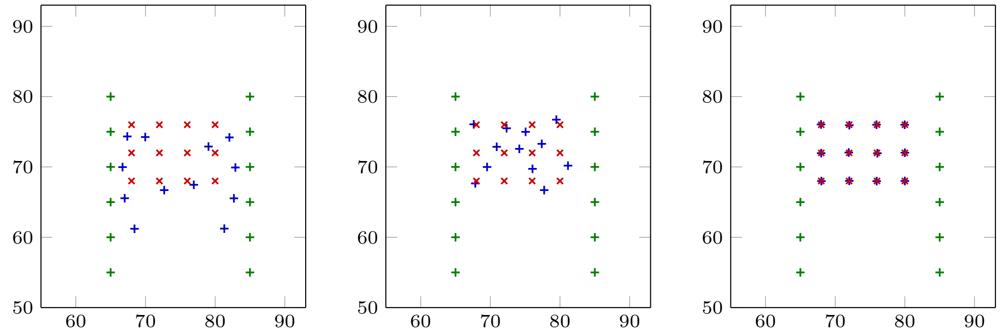


Figure 21: Progression of the obstacles position error with iterations for the reconstruction of 12 obstacles using noisy data and starting from the initial guess 1 (Figure 15(a)). The markers are only placed one iteration over four for clarity. The blue circles (···o··) indicate the SD1-LS3 method; The red crosses (···×··) indicate the SD2-LS3 method. The details of the method parameters are given Tables 6. Experiment **[12Obs.30dB.IG1]**.



(a) Iteration 2, $\kappa = 0.1$; $J = 19$; position error is 12; scaled position error is 22%.

(b) Iteration 60, $\kappa = 0.2$; $J = 0.3$; position error is 5 ; scaled position error is 10%.

(c) Iteration 525, $\kappa = 0.7$; $J = 0.25$; position error is 0.2 ; scaled position error is 0.3%.

Figure 22: Twelve obstacles position recovery using noisy data with the SD1-LS3 method and starting from initial guess 1 (Figure 15(a)). Experiment **[12Obs.30dB.IG1]**.

SD	LS	♯ frequency	$\kappa_{\max}$	♯ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	3	8	0.7	423	0.25	3.18%	0.3%	11
2	3	8	0.7	401	0.25	3.18%	0.4%	11

Table 7: Parameter used for the 12 Obstacles reconstruction using noisy data starting from initial guess 2 (Figure 15(b)). The order of FSSL method for the modeling is of 3 for frequency lower than 0.4 and order 4 for the higher frequency. SD indicates the search direction method and LS line search. Line search parameters are $\mu = 10^{-4}$ and $\eta = 0.4$ for both methods. Experiment **[12Obs.30dB.IG2]**.

8 Localization of 12 hard-scattering obstacles in random configuration

In order to test the robustness of our reconstruction method, we design an experiment with twelve hard-scattering obstacles of radius 0.5, which are now positioned randomly in the domain of interest. The domain corresponds with a section $[40, 100]_x \times [50, 100]_y$, and the obstacles are locating with their following center position:

$$\begin{array}{cccc} (75.5,\ 82.0), & (77.0,\ 71.0), & (74.0,\ 78.0), & (65.4,\ 79.9), \\ (82.0,\ 73.9), & (73.0,\ 69.0), & (70.0,\ 66.0), & (77.0,\ 65.2), \\ (67.0,\ 71.2), & (81.0,\ 68.2), & (70.0,\ 75.0), & (80.5,\ 79.8). \end{array}$$

The smallest distance between the centers of two obstacles is between 4.24 and 19.5, leading to

$$8.5 \ \leq\ \frac{\mathsf{d}}{\mathsf{r}} \ \leq\ 39\,. \tag{51}$$

The domain and the obstacles configuration is illustrated in the Figure 26, together with the initial guess for the reconstruction. The data are generated using the four angles of illumination, in a similar fashion as what has been described in the Section 7 (see Figure 14 for an illustration).

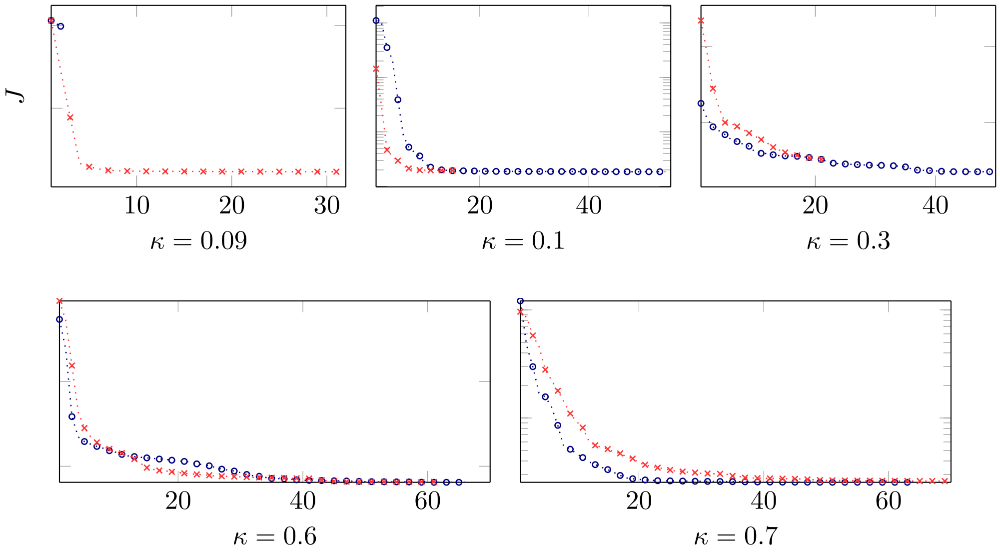


Figure 23: Evolution of the cost function depending for different frequencies for the reconstruction of 12 obstacles using noisy data and starting from the initial guess 2 (Figure 15(b)). The markers are indicated one over two iterations for clarity. The blue circles (··o··) indicate the SD1-LS3 method; The red crosses (··×··) indicate the SD2-LS3 method ; The two methods do not have the same number of iterations per frequency due to the stagnation check in the iterative algorithm. Experiment **[12Obs.30dB.IG2]**.

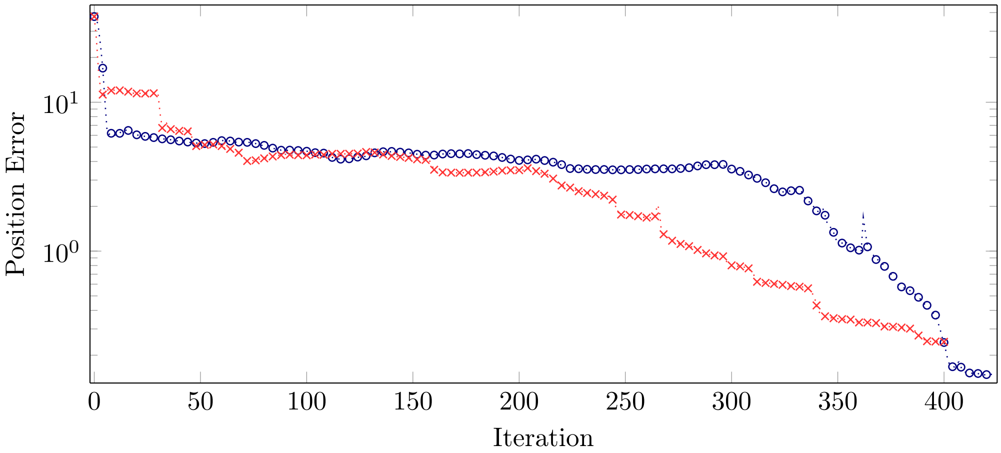
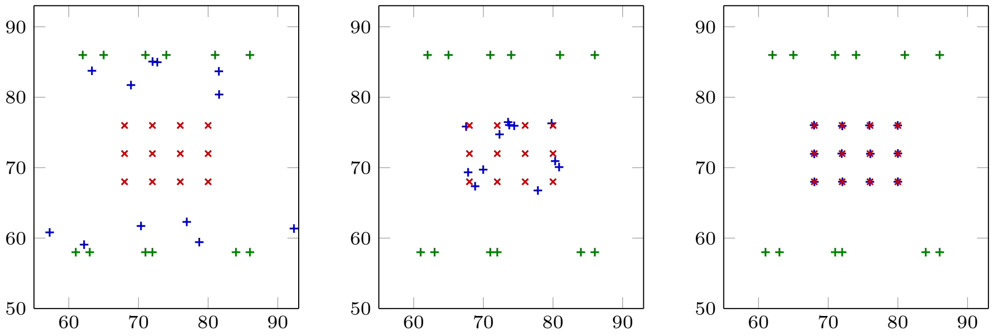


Figure 24: Progression of the obstacles position error with iterations for the reconstruction of 12 obstacles using noisy data and starting from the initial guess 2 (Figure 15(b)). The markers are only placed one iteration over four for clarity. The blue circles (··o··) indicate the SD1-LS3 method; The red crosses (··×··) indicate the SD2-LS3 method. The details of the method parameters are given Tables 7. Experiment **[12Obs.30dB.IG2]**.



(a) Iteration 2, $\kappa = 0.1$; $J = 100$; position error is 31; scaled position error is 56%.

(b) Iteration 60, $\kappa = 0.2$; $J = 0.25$; position error is 5.5 ; scaled position error is 10%.

(c) Iteration 424, $\kappa = 0.7$; $J = 0.25$; position error is 0.15 ; scaled position error is 0.25%.

Figure 25: Twelve obstacles position recovery using noisy data with the SD1-LS3 method and starting from initial guess 2 (Figure 15(b)). Experiment **[12Obs.30dB.IG2]**.

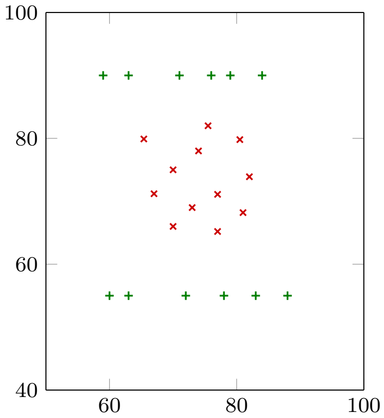


Figure 26: Configuration for the reconstruction of twelve randomly positioned obstacles (indicated in red ×). The initial guess for the reconstruction are in green (+). The position error for the initial guess if of 40, and the underlying relative error of 79.4%.

8.1 Inversion using noise-free data

We first propose a reconstruction using noise free data, with the Quasi-Newton search direction and imposing a strong Wolfe condition for line search (it corresponds to SD1-LS3 method). The set of frequencies for the reconstruction is chosen between $\kappa = 0.08$ and 0.5 for a total of seven frequencies (0.08, 0.09 and from 0.1 to 0.5 with a 0.1 step). The Table 8 provides the information of the iterative algorithm.

In Figures 28 and 27 we respectively plot the evolution of the cost function and of the obstacle position error with iterations. Compared to experiments with structured configurations, the algorithm requires more iterations, but fewer and lower frequencies to convergence to the solution. The recovery is very accurate with a final scale error in position of 0.12%, as illustrated with the visualization of the reconstruction given Figure 29.

SD	LS	♯ frequency	$\kappa_{\max}$	♯ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	3	7	0.5	507	6×10^{-7}	$4.8 \times 10^{-3}\%$	0.12%	12

Table 8: Parameter used for the 12 random obstacles reconstruction using noise-free data starting from initial guess of Figure 26. The order of FSSL method for the modeling is of 3 for frequency lower than 0.4 and order 4 for last frequency, 0.5. SD1-LS3 corresponds with Quasi-Newton search direction imposing a strong Wolfe condition for line search. Line search parameters are $\mu = 10^{-4}$ and $\eta = 0.4$. Experiment **[12Obs.Random.NoiseFree]**.

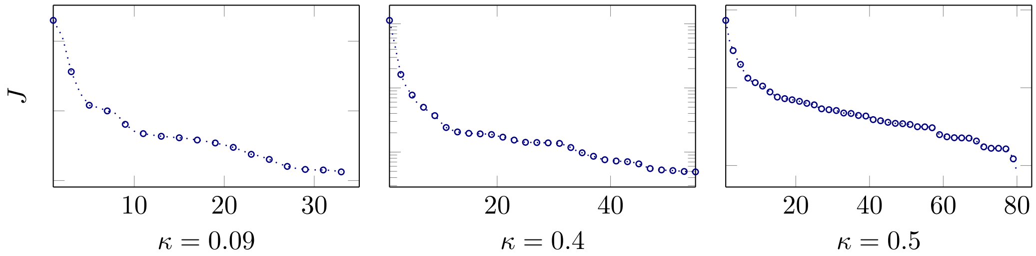


Figure 27: Evolution of the cost function for the reconstruction of twelve randomly positioned obstacles starting from the initial guess of Figure 26 and using SD1-LS3 method. The markers are indicated one over two iterations for clarity. Experiment **[12Obs.Random.NoiseFree]**.

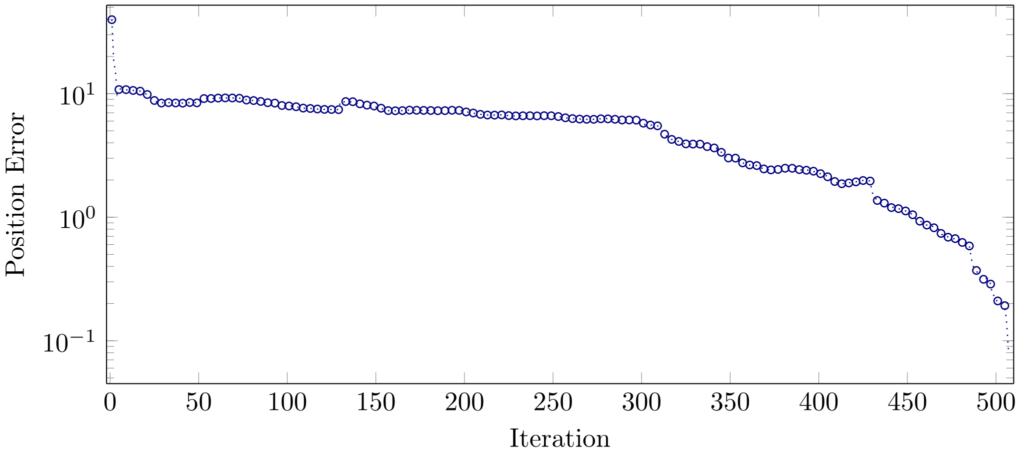
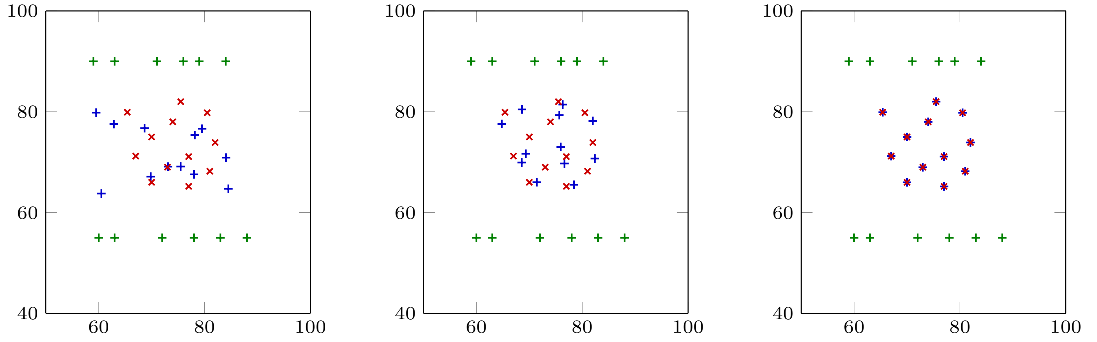


Figure 28: Progression of the obstacles position error with iterations for the reconstruction of twelve randomly positioned obstacles starting from the initial guess of Figure 26 and using SD1-LS3 method. The markers are only placed one iteration over four for clarity. Experiment **[12Obs.Random.NoiseFree]**.

8.2 Inversion using data with noise

We incorporate noise in data with 30dB signal-to-noise ratio. The L^2-norm error with the noise-free data is between 2.7% and 3.6%; the L^∞-norm is between 3.9% and 8.8%. We illustrate the data captured at the receivers location in Figure 30.

Table 9 provides the parameters of the reconstruction. Here, 10 frequencies are used, from

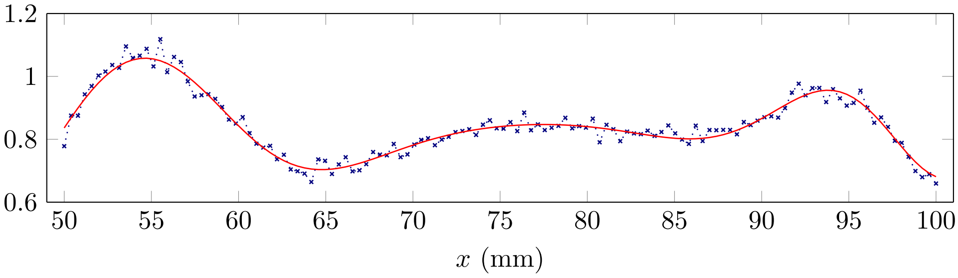


(a) Iteration 2, $\kappa = 0.09$; $J = 0.014$; position error is 15; scaled position error is 30%.

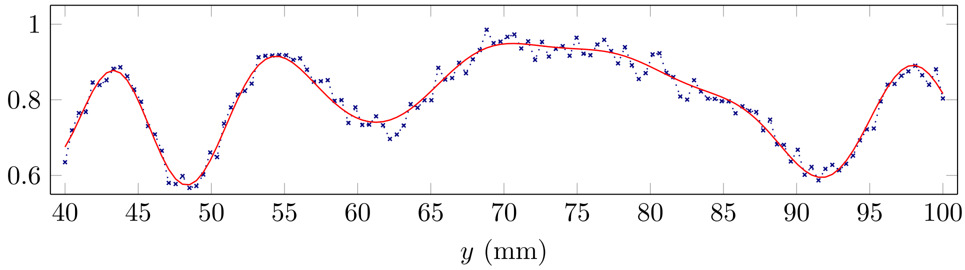
(b) Iteration 120, $\kappa = 0.2$; $J = 2.0 \times 10^{-5}$; position error is 7.5; scaled position error is 6%.

(c) Iteration 507, $\kappa = 0.5$; $J = 6.0 \times 10^{-7}$; position error is 0.06 ; scaled position error is 0.12%.

Figure 29: Twelve random obstacles position recovery using noise-free data with the SD1-LS3 method and starting from initial guess Figure 26. Initial guess are given in green (+), true positions are in red (×) and current positions in blue (+).
Experiment **[12Obs.Random.NoiseFree]**.



(a) Real part of the data for an incidence angle of 90°. Relative error between observed and noisy data is 3.3% with L^2-norm and 6.8% with L^∞ norm.



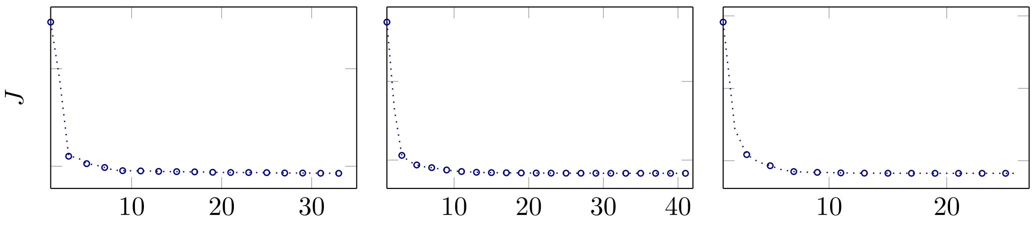
(b) Imaginary part of the data for an incidence angle of 0°. Relative error between observed and noisy data is 3% with L^2-norm and 7.1% with L^∞ norm.

Figure 30: Data obtained at the 128 receivers at frequency $\kappa = 0.8$. The red line (——) represents the synthetic data and the blue dashed line with crosses (···×···) the data incorporating 30dB noise.

$\kappa = 0.08$ to 0.8. Figure 31 shows the evolution of the cost function with iterations for three frequencies, Figure 32 shows the evolution of the obstacle position error. In this figure, we observe that despite the decrease in the cost function, the error in the obstacle position may increase along with the iteration. This is perhaps due to the randomness of the configuration. However, the final reconstruction remains very accurate, with 0.3% error in the position, as illustrated in the visualization of Figure 33.

SD	LS	♯ frequency	$\kappa_{\max}$	♯ iter.	Final J	Final Relative Error J	Final scaled position error	Run time (s)
1	3	10	0.8	317	0.24	3.05%	0.3%	8

Table 9: Parameter used for the 12 random obstacles reconstruction using noise-free data starting from initial guess of Figure 26 and using noisy data. The order of FSSL method for the modeling is of 3 for frequency lower than 0.4 and order 4 for higher frequencies. SD1-LS3 corresponds with Quasi-Newton search direction imposing a strong Wolfe condition for line search. Line search parameters are $\mu = 10^{-4}$ and $\eta = 0.4$.
Experiment **[12Obs.Random.30dB]**.



$\kappa = 0.2$ $\kappa = 0.7$ $\kappa = 0.8$

Figure 31: Evolution of the cost function for the reconstruction of twelve randomly positioned obstacles using noisy data and starting from the initial guess of Figure 26, with SD1-LS3 method. The markers are indicated one over two iterations for clarity.
Experiment **[12Obs.Random.30dB]**.

9 Conclusion

In this paper, we have implemented an iterative algorithm to locate soft or hard-scattering obstacles in either structured or random configuration inside a homogeneous medium. We follow FWI procedure to write the inverse problem as a nonlinear minimization algorithm of the cost function, which is solved using only first order information, (the value of the cost function and its gradient calculated via the Adjoint-State method). The cost of inversion procedure due to having to solve a large number of forward problems, see Remark 12, is drastically reduced by using the fast solver FSSL based upon single-layer potential. This fast solver also allows a simple implementation in FWI without the need of domain discretization.

Our experiments show that the implementation of FSSL solver in a FWI procedure gives high quality reconstruction at extremely low time cost, despite the constraints of noisy synthetic limited data (using at most four angles of radiation and at most ten frequencies), and initial guess far from the true models. For configuration of 6 obstacles, most reconstructions take less than 1 second (with the exception of that given by SD2-LS3 at 1.4s and SD2-LS1 taking 2.6s in Experiment [6Obs.23dB.IG2]). For 12 obstacles, the inversion results take around 11s (with

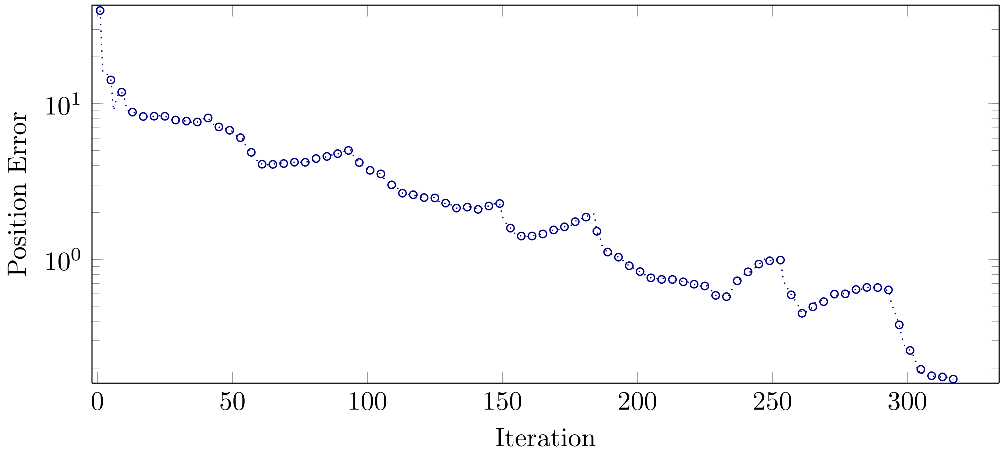
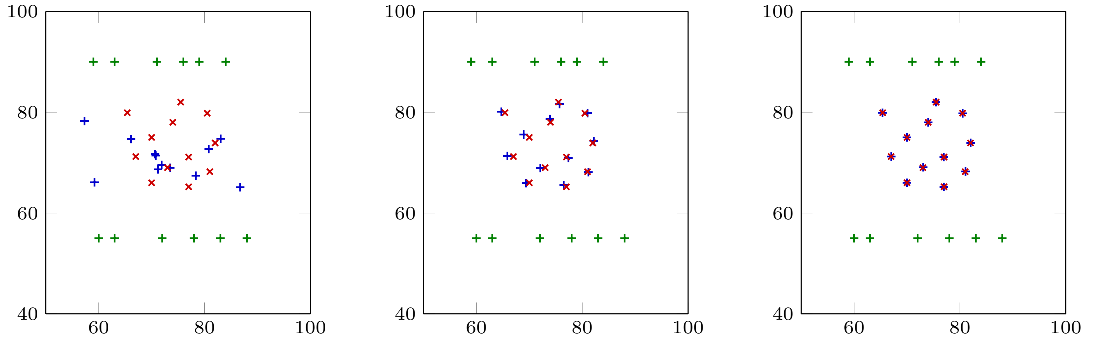


Figure 32: Progression of the obstacles position error with iterations for the reconstruction of twelve randomly positioned obstacles using noisy data and starting from the initial guess of Figure 26 with SD1-LS3 method. The markers are only placed one iteration over four for clarity. Experiment **[12Obs.Random.30dB]**.



(a) Iteration 8, $\kappa = 0.08$; $J = 0.3$; position error is 16; scaled position error is 32%.

(b) Iteration 120, $\kappa = 0.4$; $J = 0.3$; position error is 2.5; scaled position error is 5%.

(c) Iteration 317, $\kappa = 0.8$; $J = 0.17$; position error is 0.17 ; scaled position error is 0.3%.

Figure 33: Twelve random obstacles position recovery using noisy data with the SD1-LS3 method and starting from initial guess Figure 26. Initial guess are given in green (+), true positions are in red (×) and current positions in blue (+). Experiment **[12Obs.Random.30dB]**.

the exception of SD2-LS1 in Experiment [12Obs.NoiseFree.IG2] taking 23s). In addition, the reconstructions come with satisfactory precision. For both noise-free and noisy synthetic data, the reconstruction error is less than 1%, and in illustrated figures, the reconstructed positions align completely with the true positions.

We have also investigated several methods of optimization, including two search directions: Quasi-Newton BFGS (SD1) and nonlinear conjugate gradient PR+ (SD2), combined with three linesearch algorithms imposing different criteria for choosing a step length: simple backtracking (LS1), sufficient descent (LS2) and strong Wolfe conditions (LS3). In the configuration of 6 obstacles, the two search directions perform comparably well. It can happen that in some cases, simple linesearch LS1 performs very well. However, in most cases, it is overall LS3 and LS2 that give the most consistent performance. In complicated configuration with 12 obstacles, it is linesearch LS3 that shows the most reliable performance. Tested with both search directions, it gives comparable performance, however there is a slight edge towards SD1-LS3. We also note that, SD2 performs better at lower frequency, giving steeper descent in the value of the cost function, right from the start. However, at higher frequencies, it is SD1 that gives faster convergence. This means that mixed algorithms could be investigated to optimize the iterations number.

Future work This algorithm should also work in transmission problem describing fluid-fluid or fluid-elastic inclusions. In these cases, in addition to locating the inclusions, one can also try to retrieve their material parameters. In our experiments, the absence of second order information for the minimization does not prevent us from recovering accurately the solution, however, it might be interesting to investigate the convergence and calculation cost if one uses information from the Hessian. Regarding inverse techniques, it would be interesting to compare with other methods, such as MUSIC, or more interestingly couple with such a direct-imaging method to relax further the a priori information on the configuration, especially the number of obstacles. These methods can be used to generate a good initial guess, c.f. [18], and our FWI algorithm can then be used to correct the guess to the true model.

A Complex Derivation

Let $\mathbf{Z}$ be the complex vector space $\mathbf{Z} = \mathsf{Z} + \mathrm{i}\,\mathsf{Z}$. where Z is a real vector space. For $\mathbf{z} \in \mathbf{Z}$, we write

$$\mathbf{z} \,=\, \mathbf{x} \,+\, \mathrm{i}\,\mathbf{y}\;,\; x\,,y \in \mathsf{Z}\,.$$

A function g of variable $\mathbf{z} \in \mathbf{Z}$ can be considered as a function h of $\mathbf{z}$ and $\overline{\mathbf{z}}$ with these two variables treated as independent,

$$g(\mathbf{z}) \;=\; h(\mathbf{z}\,,\mathbf{w} \,=\, \overline{\mathbf{z}}) \quad ; \quad h : \mathbf{Z}_{\mathbf{z}} \times \mathbf{Z}_{\mathbf{w}} \to \mathbb{C} \quad .$$

We can also write g as a function f of the real and imaginary part of $\mathbf{z}$,

$$g(\mathbf{z}) \;=\; f(\mathbf{x},\mathbf{y}) \quad .$$

Definition 1. *If g is Fréchet differentiable with respect to $\mathbf{z}$ with $\overline{\mathbf{z}}$ treated as a constant, we denote this derivative by $\partial^{\mathsf{CR}}_{\mathbf{z}}\, g$.*

Similarly, if g is Fréchet differentiable with respect to $\overline{\mathbf{z}}$ with $\mathbf{z}$ treated as a constant, we denote this derivative by $\partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g$.

Remark 14. *With function g, h related by $g(\mathbf{z}) \;=\; h(\mathbf{z}\,,\mathbf{w} \,=\, \overline{\mathbf{z}})$. That g is Fréchet differentiable with respect to $\mathbf{z}$ with $\overline{\mathbf{z}}$ treated as a constant is equivalent to h Fréchet differentiable with respect to the first variable.*

$$\partial^{\mathsf{CR}}_{\mathbf{z}}\, g \;=\; \partial_{\mathbf{z}}\, h\,|_{\mathbf{w}=\overline{\mathbf{z}}} \;=\; \big(\partial_{\mathbf{z}}\, h\big)(\mathbf{z}\,,\,\overline{\mathbf{z}})\,. \tag{52}$$

That g is Fréchet differentiable with respect to $\overline{\mathbf{z}}$ with $\mathbf{z}$ treated as a constant is equivalent to h Fréchet differentiable with respect to the second variable,

$$\partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g \;=\; \partial_{\mathbf{w}}\, h\,|_{\mathbf{w}=\overline{\mathbf{z}}} \;=\; \big(\partial_{\mathbf{w}}\, h\big)(\mathbf{z}\,,\,\overline{\mathbf{z}}). \tag{53}$$

Proposition 3. *With f and g related as above. Suppose that*

- *f is Fréchet differentiable in $\mathbf{x}$ and $\mathbf{y}$ separately;*
- *g is Fréchet differentiable in $\mathbf{z}$ with $\overline{\mathbf{z}}$ treated as a constant, and in $\overline{\mathbf{z}}$ with $\mathbf{z}$ treated as a constant*

Then the derivatives defined in Definition 1 are equal to,

$$\partial_{\mathbf{z}}^{\mathsf{CR}}\, g \;=\; \frac{1}{2}\,(\partial_{\mathbf{x}} - \mathrm{i}\,\partial_{\mathbf{y}})\, f \quad ; \quad \partial_{\overline{\mathbf{z}}}^{\mathsf{CR}}\, g \;=\; \frac{1}{2}\,(\partial_{\mathbf{x}} + \mathrm{i}\,\partial_{\mathbf{y}})\, f \quad .$$

Proof. We consider g as a function on $\mathbf{Z} \times \mathbf{Z}$ and work with h related to g by

$$g(\mathbf{z}) \;=\; h(\mathbf{z}\,, \mathbf{w} \;=\; \overline{\mathbf{z}})\,.$$

The relation between h and f is given by

$$h\,(\mathbf{z}\,,\, \mathbf{w}) \quad = \quad \mathbf{f}\,(\mathbf{x}\,,\, \mathbf{y}) \quad \text{if} \quad \begin{pmatrix} \mathbf{z} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} 1 & \mathrm{i} \\ 1 & -\mathrm{i} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \quad .$$

Fréchet differentiating on both sides with respect to x and y

$$\begin{aligned} \big(\partial_{\mathbf{z}} h\big)\,\big(\partial_{\mathbf{x}}\mathbf{z}\big) \;+\; \big(\partial_{\mathbf{w}} h\big)\,\big(\partial_{\mathbf{x}}\mathbf{w}\big) \quad &= \quad \partial_{\mathbf{x}} f \\ \big(\partial_{\mathbf{z}} h\big)\,\big(\partial_{\mathbf{y}}\mathbf{z}\big) \;+\; \big(\partial_{\mathbf{w}} h\big)\,\big(\partial_{\mathbf{y}}\mathbf{w}\big) \quad &= \quad \partial_{\mathbf{y}} f \quad . \end{aligned}$$

Since

$$\partial_{\mathbf{x}}\mathbf{z} \,=\, 1 \quad ; \quad \partial_{\mathbf{y}}\mathbf{z} \,=\, \mathrm{i} \qquad \partial_{\mathbf{x}}\mathbf{w} \,=\, 1 \quad ; \quad \partial_{\mathbf{w}}\mathbf{z} \,=\, -i \quad ,$$

the above relation becomes

$$\begin{cases} \partial_{\mathbf{z}} h \;+\; \partial_{\mathbf{w}} h & = \partial_{\mathbf{x}} f \quad ; \\ \mathrm{i}\,\partial_{\mathbf{z}} h \;-\; \mathrm{i}\,\partial_{\mathbf{w}} h & = \partial_{\mathbf{y}} f \quad \Rightarrow \; \partial_{\mathbf{z}} h - \partial_{\mathbf{w}} h \;=\; -\,\mathrm{i}\,\partial_{\mathbf{y}} f \end{cases} \Longrightarrow \begin{cases} \partial_{\mathbf{z}} h \;=\; \frac{1}{2}\,(\partial_{\mathbf{x}} - \mathrm{i}\,\partial_{\mathbf{y}}) f \\ \partial_{\mathbf{w}} h \;=\; \frac{1}{2}\,(\partial_{\mathbf{x}} + \mathrm{i}\;\partial_{\mathbf{y}}) f \end{cases} \quad .$$

We use (52) and (53) to go back to the CR-derivatives of g. □

As a corollary, if g and thus f is real-valued then

$$\partial_{\mathbf{x}} f \;=\; \overline{\partial_{\mathbf{x}} f} \quad ; \quad \partial_{\mathbf{y}} f \;=\; \overline{\partial_{\mathbf{y}} f}\,,$$

and we obtain

$$2\,\overline{\partial_{\mathbf{z}}^{\mathsf{CR}} g} \;=\; \overline{\partial_{\mathbf{x}} f - \mathrm{i}\,\partial_{\mathbf{y}} f} \;=\; \partial_{\mathbf{x}} f \,+\, \mathrm{i}\,\partial_{\mathbf{y}} f \;=\; 2\,\partial_{\overline{\mathbf{z}}}^{\mathsf{CR}} g \quad . \tag{54}$$

Proposition 4 (Chain Rule)**.** *1. Consider the following real-valued functions, h, g and f related by*

$$h(p) \;=\; g\big(\mathbf{z}(p)\big) \;=\; f(\mathbf{x}\,,\, \mathbf{y}) \quad .$$

Suppose

- *h is Fréchet differentiable in p,*
- *f is Fréchet differentiable in $\mathbf{x}$ and $\mathbf{y}$,*
- *$\mathbf{x}$ and $\mathbf{y}$ Fréchet differentiable with respect to p.*
- *g is Fréchet differentiable in $\mathbf{z}$ with $\overline{\mathbf{z}}$ kept constant, and in $\overline{\mathbf{z}}$ with $\mathbf{z}$ kept constant.*

Then

$$(\partial_p h) \;=\; 2\,\mathrm{Re}\big(\partial^{\mathsf{CR}}_{\mathbf{z}}\, g\;\partial_p\mathbf{z}\big) \;=\; 2\,\mathrm{Re}\big(\partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g\;\partial_p\overline{\mathbf{z}}\big)\,. \tag{55}$$

2. For the complex-valued functions, h, g related by

$$h(p) \;=\; g\big(\mathbf{z}(p)\big)$$

with the same assumption for h and g as above. Then

$$(\partial_p h) \;=\; \partial^{\mathsf{CR}}_{\mathbf{z}}\, g\;\partial_p\mathbf{z} \;+\; \overline{\partial^{\mathsf{CR}}_{\mathbf{z}}\,\overline{g}\;\partial_p\mathbf{z}} \;=\; \partial^{\mathsf{CR}}_{\mathbf{z}}\, g\;\partial_p\mathbf{z} \;+\; \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g\;\partial_p\overline{\mathbf{z}}\,. \tag{56}$$

If the definition of g does not involve $\overline{\mathbf{z}}$, then

$$(\partial_p h) \;=\; \partial^{\mathsf{CR}}_{\mathbf{z}}\, g\;\partial_p\mathbf{z} \quad .$$

Proof. We have the following identities,

$$\begin{pmatrix}\partial^{\mathsf{CR}}_{\mathbf{z}}\\ \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\end{pmatrix} g = \frac{1}{2}\begin{pmatrix}1 & -\mathrm{i}\\ 1 & \mathrm{i}\end{pmatrix}\begin{pmatrix}\partial_{\mathbf{x}}\\ \partial_{\mathbf{y}}\end{pmatrix} f \quad ; \quad \begin{pmatrix}\partial_{\mathbf{x}}\\ \partial_{\mathbf{y}}\end{pmatrix} f = \begin{pmatrix}1 & 1\\ \mathrm{i} & -\mathrm{i}\end{pmatrix}\begin{pmatrix}\partial^{\mathsf{CR}}_{\mathbf{z}}\\ \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\end{pmatrix} g$$

$$\partial_p\begin{pmatrix}\mathbf{z}\\ \overline{\mathbf{z}}\end{pmatrix} = \begin{pmatrix}1 & \mathrm{i}\\ 1 & -i\end{pmatrix}\partial_p\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \quad ; \quad \partial_p\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1\\ -\mathrm{i} & i\end{pmatrix}\partial_p\begin{pmatrix}\mathbf{z}\\ \overline{\mathbf{z}}\end{pmatrix}$$

Part 1 Use the above identities to calculate $\partial_p h$,

$$\begin{aligned}(\partial_p h) &= (\partial_{\mathbf{x}} f)\,\partial_p\mathbf{x} \quad + \quad (\partial_{\mathbf{y}} f)\,\partial_p\mathbf{y}\\ &= \big(\partial^{\mathsf{CR}}_{\mathbf{z}}\, g + \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g\big)\,\tfrac{1}{2}\,\big(\partial_p\,\mathbf{z} + \partial_p\overline{\mathbf{z}}\big) - \quad \big(\partial^{\mathsf{CR}}_{\mathbf{z}} g - \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}} g\big)\;\tfrac{1}{2}\;\big(-\partial_p\mathbf{z} + \partial_p\overline{\mathbf{z}}\big)\\ &= \partial^{\mathsf{CR}}_{\mathbf{z}}\, g\,\partial_p\,\mathbf{z} \quad + \quad \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g\,\partial_p\overline{\mathbf{z}}\\ &= 2\,\mathrm{Re}\big(\partial^{\mathsf{CR}}_{\mathbf{z}} g\;\partial_p\mathbf{z}\big) \quad = \quad 2\,\mathrm{Re}\big(\partial^{\mathsf{CR}}_{\overline{\mathbf{z}}} g\;\partial_p\overline{\mathbf{z}}\big) \quad .\end{aligned}$$

The last equality comes from identity (54) when g is real-valued, and p is a real variable, i.e

$$\partial^{\mathsf{CR}}_{\mathbf{z}}\, g \;=\; \overline{\partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g} \quad ; \quad \partial_p\,\overline{\mathbf{z}} = \overline{\partial_p\,\mathbf{z}} \quad .$$

Part 2 Applying the above result to the real and imaginary part of a complex-valued function h

$$h(x,y) = h_1(x,y) + \mathrm{i}\,h_2(x,y) = g(\mathbf{z}) = g_1(\mathbf{z}) + \mathrm{i}\,g_2(\mathbf{z})$$

then we obtain

$$
\begin{aligned}
\partial_p h \;&=\; \partial_p\, h_1 \;+\; \mathrm{i}\,\partial_p h_2 \\
&=\; 2\ \mathrm{Re}\,\Big(\partial^{\mathsf{CR}}_{\mathbf{z}} g_1\ \partial_p \mathbf{z}\Big) \;+\; \mathrm{i}\,2\ \mathrm{Re}\,\Big(\partial^{\mathsf{CR}}_{\mathbf{z}} g_2\ \partial_p \mathbf{z}\Big) \\
&=\; \partial^{\mathsf{CR}}_{\mathbf{z}} g_1\ \partial_p \mathbf{z} \;+\; \overline{\partial^{\mathsf{CR}}_{\mathbf{z}} g_1\ \partial_p \mathbf{z}} \;+\; \mathrm{i}\,\Big(\,\partial^{\mathsf{CR}}_{\mathbf{z}} g_2\ \partial_p \mathbf{z} \;+\; \overline{\partial^{\mathsf{CR}}_{\mathbf{z}} g_2\ \partial_p \mathbf{z}} \\
&=\; \partial^{\mathsf{CR}}_{\mathbf{z}} g_1\ \partial_p \mathbf{z} \;+\; \mathrm{i}\ \partial^{\mathsf{CR}}_{\mathbf{z}} g_2\ \partial_p \mathbf{z} \;+\; \overline{\partial^{\mathsf{CR}}_{\mathbf{z}} g_1\ \partial_p \mathbf{z} \;-\; \mathrm{i}\,\partial^{\mathsf{CR}}_{\mathbf{z}} g_2\ \partial_p \mathbf{z}} \\
&=\; \partial^{\mathsf{CR}}_{\mathbf{z}} g\ \partial_p \mathbf{z} \;+\; \overline{\partial^{\mathsf{CR}}_{\mathbf{z}} \overline{g}\ \partial_p \mathbf{z}} \quad .
\end{aligned}
$$

If g is independent of $\overline{\mathbf{z}}$ and thus $\overline{g}$ of $\mathbf{z}$, $\partial^{\mathsf{CR}}_{\mathbf{z}}\,\overline{g} \;=\; \partial^{\mathsf{CR}}_{\overline{\mathbf{z}}}\, g \;=\; 0$. In this case, for h (Fréchet)-differential in p such that g is independent of $\mathbf{z}$ (whose definition includes only $\mathbf{z}$ and not $\overline{\mathbf{z}}$) then

$$
\partial_p\, h \;=\; \partial^{\mathsf{CR}}_{\mathbf{z}} g\ \partial_p \mathbf{z} \quad .
$$

□

Application Consider the real-valued function ,

$$
J(u) \;=\; \frac{1}{2}\,\Big\|\mathcal{R}\,u - \mathbf{d}_{\mathrm{obs}}\Big\|^2 \quad ; \quad \hat{J}(p) \;=\; \frac{1}{2}\,\Big\|\mathcal{R}\,u(p) - \mathbf{d}_{\mathrm{obs}}\Big\|^2
$$

Here, $\mathcal{R}$ is a linear operator evaluation function to give column row vector of size N_{rec}, see (16). Using the fact that $\overline{\mathcal{R}\,u} \;=\; \mathcal{R}\,\overline{u}$, we can write J as

$$
J \;=\; \frac{1}{2}\Big(\mathcal{R}\,u - \mathbf{d}_{\mathrm{obs}}\Big)^{\star}\Big(\mathcal{R}\,u - \mathbf{d}_{\mathrm{obs}}\Big) \;=\; \frac{1}{2}\Big(\mathcal{R}\,\overline{u} - \overline{\mathbf{d}_{\mathrm{obs}}}\Big)^{t}\Big(\mathcal{R}\,u - \mathbf{d}_{\mathrm{obs}}\Big) \quad ,
$$

By Definition 1, we have

$$
\partial^{\mathsf{CR}}_{u}\, J \quad = \quad \tfrac{1}{2}\ (\mathcal{R}u - \mathbf{d}_{\mathrm{obs}})^{\star}\ \mathcal{R} \quad .
$$

By (55) in Prop 4, we have

$$
\partial_p\ \hat{J} \;=\; 2\,\mathrm{Re}\Big(\partial^{\mathsf{CR}}_{u} J \quad \underbrace{\partial_p u}_{1\times N_{\mathrm{par}}} \quad\Big) = 2\,\mathrm{Re}\Big(\big(\mathcal{R}u - \mathbf{d}_{\mathrm{obs}}\big)^{\star} \quad \underbrace{\mathcal{R}\ \partial_p u}_{N_{\mathrm{rec}}\times N_{\mathrm{par}}} \quad\Big) \quad . \tag{57}
$$

B Explicit computation of derivatives

Inverse Problem 1 : for Dirichlet scattering, known number of obstacles with known radius. Goal : retrieve the position of the obstacles. Thus, in this case, the parameters are the positions of the obstacles, i.e.

$$
\boldsymbol{p} = (\mathbf{x}_1\,,\,\ldots\,,\,\mathbf{x}_{N_{\mathrm{Obs}}}) \;\in \mathbb{R}^{N_{\mathrm{par}}} \quad ; \quad \mathbf{x}_I = (\mathbf{x}_{I,1}\,,\,\mathbf{x}_{I,2})\,,\, N_{\mathrm{par}} = 2N_{\mathrm{Obs}} \quad .
$$

Notation For a scalar-valued function f, we write

$$
\nabla_{\mathbf{x}_I} f = (\partial_{\mathbf{x}_{I,1}} f, \partial_{\mathbf{x}_{I,2}} f)^t \quad ; \quad \partial_{\mathbf{x}_I} f = (\partial_{\mathbf{x}_{I,1}} f, \partial_{\mathbf{x}_{I,2}} f) \quad .
$$

$$
\nabla_{\boldsymbol{p}} \;=\; \begin{pmatrix} \nabla_{\mathbf{x}_1} \\ \vdots \\ \nabla_{\mathbf{x}_{N_{\mathrm{Obs}}}} \end{pmatrix} \quad ; \quad \partial_{\boldsymbol{p}} \;=\; \Big(\partial_{\mathbf{x}_{1,1}}\ \partial_{\mathbf{x}_{1,2}}\ \cdots\ \partial_{\mathbf{x}_{N_{\mathrm{Obs}},1}}\ \partial_{\mathbf{x}_{N_{\mathrm{Obs}},2}}\Big)
$$

B.1 Derivatives of the multiple-scattering matrix

Here we calculate $\partial_{\boldsymbol{p}}\mathbf{A}$ and matrix W . Recall $\mathsf{W}(\boldsymbol{p})$ is matrix of size $N \times N_{\text{par}}$, with components given by,

$$\left(\underbrace{\sum_{\tilde{J}=1}^{N_{\text{Obs}}} \sum_{m=-\mathbf{m}}^{\mathbf{m}} \left(\partial_{\boldsymbol{p}}\, \mathbf{A}_{I\tilde{J};lm}\right) \mathsf{S}_{\tilde{J}m}}_{\text{the } (I,l)\text{-th row of } \mathsf{W}} \right)_{\substack{1\le I\le N_{\text{Obs}} \\ -\mathbf{m}\le l\le \mathbf{m}}} .$$

Decomposition of multiple-scattering matrix A We decompose the coefficient matrix as

$$\mathbf{A} \quad = \quad \mathbf{D}_1\,\mathbf{D}_2 \;+\; \mathbf{D}_1\,\mathbf{T}\,\mathbf{D}_3 \quad . \tag{58}$$

The definitions of the diagonal matrices are given as follows. The diagonal matrices do not depend on the position of the obstacles. The diagonal matrix $\mathbf{D}_1$ depends on the radius of the obstacles r_I and the external wavenumber κ.

$$\big(\mathbf{D}_1\big)_{I,ll} = \mathrm{i}\,\pi\,\mathsf{r}_J\;\mathrm{J}_m(\kappa\,\mathsf{r}_J)$$

The diagonal matrix $\mathbf{D}_2$ depends on the type of boundary conditions (and hence the parameters associated with these boundary conditions), the radius of the obstacles r_I and the external wavenumber κ.

$$\big(\mathbf{D}_2\big)_{I,ll} = \begin{cases} \mathrm{H}_l^{(1)}(\kappa\,\mathsf{r}_I) & , \quad \text{soft-scattering} \\ \kappa\,\mathrm{H}_l^{(1)\prime}(\kappa\,\mathsf{r}_I) & , \quad \text{hard-scattering} \\ \mathrm{i}\,\lambda\,\mathrm{H}_l^{(1)}(\kappa\,\mathsf{r}_I) \;+\; \kappa\,\mathrm{H}_l^{(1)\prime}(\kappa\,\mathsf{r}_I) & , \quad \text{Impedance scattering} \end{cases} .$$

$$\big(\mathbf{D}_3\big)_{I,ll} = \begin{cases} \mathrm{J}_l(\kappa\,\mathsf{r}_I) & , \quad \text{soft-scattering} \\ \kappa\,J_l'(\kappa\,\mathsf{r}_I) & , \quad \text{hard-scattering} \\ \mathrm{i}\,\lambda\,\mathrm{J}_l(\kappa\,\mathsf{r}_I) \;+\; \kappa\,J_l'(\kappa\,\mathsf{r}_I) & , \quad \text{Impedance scattering} \end{cases} .$$

The matrix $\mathbf{T}$ has zero diagonal blocks, and its off-diagonal blocks being Toeplitz, given by

$$\mathbf{T}_{IJ} = O_{(2\mathbf{m}+1)\times(2\mathbf{m}+1)} \;, I = J \quad ; \quad (\mathbf{T}_{IJ})_{lm} = \mathrm{H}_{m-l}^{(1)}\,(\kappa\,\mathsf{d}_{IJ})\; e^{i(m-l)\theta_{JI}} \;\; I \neq J \;.$$

This matrix depends on the relative positions of the obstacles among themselves.

In short, in the decomposition of $\mathbf{A}$, the matrices $\mathbf{D}_i$ do not depend on the position of the obstacles $\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\text{Obs}}}$, i.e on p. Thus,

$$\partial_{\mathbf{x}_{I,i}}\mathbf{A} = \mathbf{D}_1\,\big(\partial_{\mathbf{x}_{I,i}}\mathbf{T}\big)\,\mathbf{D}_3 \quad .$$

Equivalently,

$$\partial_{\boldsymbol{p}}\;\mathbf{A}_{I\tilde{J};lm}(\boldsymbol{p}) \quad = \quad (\mathbf{D}_3)_{Il}\,\left(\partial_{\boldsymbol{p}}\mathbf{T}_{I\tilde{J};lm}\right)\,(\mathbf{D}_1)_{\tilde{J}m} \quad .$$

Returning to W,

$$\text{Row}_{Il}(\mathsf{W}) \;=\; \left(\quad \underbrace{\sum_{m=-\mathbf{m}}^{\mathbf{m}} \sum_{\tilde{J}=1}^{N_{\text{Obs}}} (\mathbf{D}_3)_{Il}\,\left(\partial_{\mathbf{x}_J}\mathbf{T}_{I\tilde{J};lm}\right)\,(\mathbf{D}_1)_{\tilde{J}m}\;\mathsf{S}_{\tilde{J}m} \quad }_{\text{row vector of size 2}} \right)_{1\le J\le N_{\text{Obs}}} .$$

Since $\partial_{\mathbf{x}_{J,i}} \mathbf{T}_{I\tilde{J}}$, with $i = 1, 2$, is all zero except for $\begin{cases} (J = I \text{ or } \tilde{J} = J) \\ \text{and } \tilde{J} \neq I \end{cases}$, the term in the parentheses can be simplified to

$$\Big[\mathrm{Row}_{Il}(\mathsf{W})\Big](2J-1:2J) \ = (\mathbf{D}_3)_{Il} \times \begin{cases} \sum_{m=-\mathbf{m}}^{\mathbf{m}} \sum_{\substack{\tilde{J}=1 \\ \tilde{J}\neq I}}^{N_{\mathrm{Obs}}} \Big(\partial_{\mathbf{x}_I} \mathbf{T}_{I\tilde{J};lm}\Big) \, (\mathbf{D}_1)_{\tilde{J}m} \, \mathsf{S}_{\tilde{J}m} & , \ J = I \\ \sum_{m=-\mathbf{m}}^{\mathbf{m}} \Big(\partial_{\mathbf{x}_J} \mathbf{T}_{IJ;lm}\Big) \, (\mathbf{D}_1)_{Jm} \, \mathsf{S}_{Jm} & , \ J \neq I \end{cases}$$

We recall from (63) and (64) that

$$\nabla_{\mathbf{x}_I} (\mathbf{T}_{I\tilde{J}})_{lm} = e^{\mathrm{i}\,(m-l)\,\theta_{\tilde{J}I}} \left[\kappa \, \mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{I\tilde{J}}) \begin{pmatrix} \cos\theta_{\tilde{J}I} \\ \sin\theta_{\tilde{J}I} \end{pmatrix} + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{I\tilde{J}})}{\mathsf{d}_{I\tilde{J}}} \begin{pmatrix} -\sin\theta_{\tilde{J}I} \\ \cos\theta_{\tilde{J}I} \end{pmatrix} \right]$$

$$\nabla_{\mathbf{x}_J} (\mathbf{T}_{IJ})_{lm} = e^{\mathrm{i}\,(m-l)\,\theta_{IJ}} \left[\kappa \, \mathrm{H}^{(1)\prime}_{l-m}\,(\kappa\,\mathsf{d}_{IJ}) \begin{pmatrix} \cos\theta_{IJ} \\ \sin\theta_{IJ} \end{pmatrix} + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{l-m}\,(\kappa\,\mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} -\sin\theta_{IJ} \\ \cos\theta_{IJ} \end{pmatrix} \right] .$$

For $J = I$

$$\begin{aligned} \Big[\mathrm{Row}_{Il}(\mathsf{W})\Big](2I-1:2I) = \mathbf{D}_{3;Il} \sum_{m=-\mathbf{m}}^{\mathbf{m}} \sum_{\substack{\tilde{J}=1 \\ \tilde{J}\neq I}}^{N_{\mathrm{Obs}}} e^{\mathrm{i}\,(m-l)\,\theta_{\tilde{J}I}} \Big[\kappa \, \mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{I\tilde{J}}) \begin{pmatrix} \cos\theta_{\tilde{J}I} \\ \sin\theta_{\tilde{J}I} \end{pmatrix}^t \\ + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{I\tilde{J}})}{\mathsf{d}_{I\tilde{J}}} \begin{pmatrix} -\sin\theta_{\tilde{J}I} \\ \cos\theta_{\tilde{J}I} \end{pmatrix}^t \Big] \, \mathbf{D}_{1;\tilde{J}m} \, \mathsf{S}_{\tilde{J}m} \, . \end{aligned}$$

For $J \neq I$ then

$$\begin{aligned} \Big[\mathrm{Row}_{Il}(\mathsf{W})\Big](2J-1:2J) = \mathbf{D}_{3;Il} \sum_{m=-\mathbf{m}}^{\mathbf{m}} e^{\mathrm{i}\,(m-l)\,\theta_{IJ}} \Big[\kappa \, \mathrm{H}^{(1)\prime}_{l-m}\,(\kappa\,\mathsf{d}_{IJ}) \begin{pmatrix} \cos\theta_{IJ} \\ \sin\theta_{IJ} \end{pmatrix}^t \\ + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{l-m}\,(\kappa\,\mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} -\sin\theta_{IJ} \\ \cos\theta_{IJ} \end{pmatrix}^t \Big] \, \mathbf{D}_{1;Jm} \, \mathsf{S}_{Jm}. \end{aligned}$$

For $V \in \mathbb{C}^N$, consider $V^t \, \mathsf{W}$, which is a row vector of size N_{par}, the I-th block of $V^t \, \mathsf{W}$ is given by

$$
\begin{aligned}
\left(V^t\,\mathbf{W}\right)_{2I-1:2I} &= \sum_{\tilde{I}=1}^{N_{\text{Obs}}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{\tilde{I}l}\,\mathbf{W}_{\tilde{I}l,2I-1:2I} \\
&= \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{Il}\,\mathbf{W}_{Il,2I-1:2I} \;+\; \sum_{\substack{\tilde{I}=1\\ \tilde{I}\neq I}}^{N_{\text{Obs}}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{\tilde{I}l}\,\mathbf{W}_{\tilde{I}l,2I-1:2I} \\
&= \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{Il}\,\mathbf{D}_{3;Il} \sum_{m=-\mathbf{m}}^{\mathbf{m}} \sum_{\substack{\tilde{J}=1\\ \tilde{J}\neq I}}^{N_{\text{Obs}}} e^{\mathrm{i}\,(m-l)\,\theta_{\tilde{J}I}} \Big[\kappa\,\mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{I\tilde{J}}) \begin{pmatrix}\cos\theta_{\tilde{J}I}\\ \sin\theta_{\tilde{J}I}\end{pmatrix}^t \\
&\qquad\qquad + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{I\tilde{J}})}{\mathsf{d}_{I\tilde{J}}} \begin{pmatrix}-\sin\theta_{\tilde{J}I}\\ \cos\theta_{\tilde{J}I}\end{pmatrix}^t \Big]\mathbf{D}_{1;\tilde{J}m}\,\mathsf{S}_{\tilde{J}m} \\
&\quad + \sum_{\substack{\tilde{I}=1\\ \tilde{I}\neq I}}^{N_{\text{Obs}}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{\tilde{I}l}\,\mathbf{D}_{3;\tilde{I}l} \sum_{m=-\mathbf{m}}^{\mathbf{m}} e^{\mathrm{i}\,(m-l)\,\theta_{\tilde{I}I}} \Big[\kappa\,\mathrm{H}^{(1)\prime}_{l-m}\,(\kappa\,\mathsf{d}_{\tilde{I}I}) \begin{pmatrix}\cos\theta_{\tilde{I}I}\\ \sin\theta_{\tilde{I}I}\end{pmatrix}^t \\
&\qquad\qquad + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{l-m}\,(\kappa\,\mathsf{d}_{\tilde{I}I})}{\mathsf{d}_{\tilde{I}I}} \begin{pmatrix}-\sin\theta_{\tilde{I}I}\\ \cos\theta_{\tilde{I}I}\end{pmatrix}^t \Big]\mathbf{D}_{1;Im}\,\mathsf{S}_{Im}\,.
\end{aligned}
$$

We next change the notation of $\tilde{J}$ to J in the first two sums, and $\tilde{I}$ to J in the last sum.

$$
\begin{aligned}
&= \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{Il}\,\mathbf{D}_{3;Il} \sum_{m=-\mathbf{m}}^{\mathbf{m}} \sum_{\substack{J=1\\ J\neq I}}^{N_{\text{Obs}}} e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \Big[\kappa\,\mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \begin{pmatrix}\cos\theta_{JI}\\ \sin\theta_{JI}\end{pmatrix} \\
&\qquad\qquad + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix}-\sin\theta_{JI}\\ \cos\theta_{JI}\end{pmatrix} \Big]\mathbf{D}_{1;Jm}\,\mathsf{S}_{Jm} \\
&\quad + \sum_{\substack{J=1\\ J\neq I}}^{N_{\text{Obs}}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{Jl}\,\mathbf{D}_{3;Jl} \sum_{m=-\mathbf{m}}^{\mathbf{m}} e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \Big[\kappa\,\mathrm{H}^{(1)\prime}_{l-m}\,(\kappa\,\mathsf{d}_{JI}) \begin{pmatrix}\cos\theta_{JI}\\ \sin\theta_{JI}\end{pmatrix}^t \\
&\qquad\qquad + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{l-m}\,(\kappa\,\mathsf{d}_{JI})}{\mathsf{d}_{JI}} \begin{pmatrix}-\sin\theta_{JI}\\ \cos\theta_{JI}\end{pmatrix}^t \Big]\mathbf{D}_{1;Im}\,\mathsf{S}_{Im}.
\end{aligned}
$$

Next switch l and m in the second term

$$
\begin{aligned}
&= \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{Il}\,\mathbf{D}_{3;Il} \sum_{m=-\mathbf{m}}^{\mathbf{m}} \sum_{\substack{J=1\\ J\neq I}}^{N_{\text{Obs}}} e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \Big[\kappa\,\mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \begin{pmatrix}\cos\theta_{JI}\\ \sin\theta_{JI}\end{pmatrix} \\
&\qquad\qquad + \frac{\mathrm{i}\,(m-l)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix}-\sin\theta_{JI}\\ \cos\theta_{JI}\end{pmatrix} \Big]\mathbf{D}_{1;Jm}\,\mathsf{S}_{Jm} \\
&\quad + \sum_{\substack{J=1\\ J\neq I}}^{N_{\text{Obs}}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} V_{Jm}\,\mathbf{D}_{3;Jm} \sum_{m=-\mathbf{m}}^{\mathbf{m}} e^{\mathrm{i}\,(l-m)\,\theta_{JI}} \Big[\kappa\,\mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{JI}) \begin{pmatrix}\cos\theta_{JI}\\ \sin\theta_{JI}\end{pmatrix}^t \\
&\qquad\qquad + \frac{\mathrm{i}\,(l-m)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{JI})}{\mathsf{d}_{JI}} \begin{pmatrix}-\sin\theta_{JI}\\ \cos\theta_{JI}\end{pmatrix}^t \Big]\mathbf{D}_{1;Il}\,\mathsf{S}_{Il} \quad .
\end{aligned}
$$

As a conclusion, for general column vector W of size N and independent of $\boldsymbol{p}$.

$$W^t \, \mathbf{W}(\boldsymbol{p}) \;=\; \Big(\, \partial_{\boldsymbol{p}} \, \Big[W^t \mathbf{A}(\boldsymbol{p}) \, V \, \Big] \Big) \, \Big|_{V = \mathsf{S}(\boldsymbol{p})} \, . \tag{59}$$

where for general column vector W, V of size N and independent of $\boldsymbol{p}$, the quantity $\partial_{\boldsymbol{p}} \; W^t \mathbf{A}(\boldsymbol{p}) \, V$ is a row vector of size N_{par} and is given by,

$$\begin{aligned}
&\big(\, \partial_{\boldsymbol{p}} \; W^t \mathbf{A}(\boldsymbol{p}) \, V \; \big)(2I-1:2I) \\
&= \sum_{J=1}^{N_{\text{Obs}}} \left[\sum_{l,m=-\mathbf{m}}^{\mathbf{m}} W_{Il} \; \boldsymbol{\alpha}_{Il} \; \mathcal{M}^{(1)}_{IJ,lm} \; \boldsymbol{\beta}_{Jm} \; V_{Jm} \; + \; W_{Jm} \; \boldsymbol{\alpha}_{Jm} \; \mathcal{M}^{(2)}_{IJ;lm} \; \boldsymbol{\beta}_{Il} \; V_{Il} \right] \begin{pmatrix} \cos\theta_{JI} \\ \sin\theta_{JI} \end{pmatrix}^t \\
&+ \sum_{J=1}^{N_{\text{Obs}}} \left[\sum_{l,m=-\mathbf{m}}^{\mathbf{m}} W_{Il} \; \boldsymbol{\alpha}_{Il} \; \mathcal{M}^{(3)}_{IJ,lm} \; \boldsymbol{\beta}_{Jm} \; V_{Jm} \; + \; W_{Jm} \; \boldsymbol{\alpha}_{Jm} \; \mathcal{M}^{(4)}_{IJ,lm} \; \boldsymbol{\beta}_{Il} \; V_{Il} \right] \begin{pmatrix} -\sin\theta_{JI} \\ \cos\theta_{JI} \end{pmatrix}^t
\end{aligned} \tag{60}$$

Here, we have defined block matrix $\mathcal{M}^{(1)}_{IJ}$, $\mathcal{M}^{(2)}_{IJ}$, $\mathcal{M}^{(3)}_{IJ}$, $\mathcal{M}^{(4)}_{IJ}$, and matrix $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

$$\begin{aligned}
I = J \quad &: \quad \mathcal{M}^{(1)}_{IJ} \;=\; \mathcal{M}^{(2)}_{IJ} \;=\; \mathcal{M}^{(3)}_{IJ} \;=\; \mathcal{M}^{(4)}_{IJ} \;=\; \mathbf{0}_{(2\mathbf{m}+1)\times(2\mathbf{m}+1)} \\
I \neq J \quad &: \quad \mathcal{M}^{(1)}_{IJ,lm} \;=\; \mathrm{i}\,\pi\,\mathsf{r}_J \;\; e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \;\; \kappa\,\mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \;\; \mathrm{J}_m(\kappa\,\mathsf{r}_J) \quad ; \\
& \mathcal{M}^{(2)}_{IJ,lm} \;=\; \mathrm{i}\,\pi\,\mathsf{r}_I \;\; e^{\mathrm{i}\,(l-m)\,\theta_{JI}} \;\; \kappa\,\mathrm{H}^{(1)\prime}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \;\; \mathrm{J}_m(\kappa\,\mathsf{r}_J) \quad ; \\
& \mathcal{M}^{(3)}_{IJ,lm} \;=\; -\,\pi\,\mathsf{r}_J\,e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \;\; \frac{(m-l)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \quad ; \\
& \mathcal{M}^{(4)}_{IJ,lm} \;=\; -\,\pi\,\mathsf{r}_J\,e^{\mathrm{i}\,(l-m)\,\theta_{JI}} \;\; \frac{(l-m)\,\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \, .
\end{aligned} \tag{61}$$

and

$$\boldsymbol{\alpha}_{Il} = \begin{cases} \mathrm{J}_l(\kappa\,\mathsf{r}_I) & \text{Dirichlet BC} \\ \kappa\,J_l'(\kappa\,\mathsf{r}_I) & \text{Neumann BC} \end{cases} \quad ; \quad \boldsymbol{\beta}_{Jm} = \mathrm{J}_m(\kappa\,\mathsf{r}_J)\,. \tag{62}$$

Remark 15. *Expression* (60) *is coded in subroutine* `GradCostTerm3_v2`*. The matrices* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$ *correspond to* `temp1a`*,* `temp1b`*,* `temp2a`*,*`temp2b`*. Component* $\big(\partial_{\boldsymbol{p}} \; W^t \mathbf{A}(\boldsymbol{p}) \, V\big)(2I-1)$ *is given by* `TempX` *and* $\big(\partial_{\boldsymbol{p}} \; W^t \mathbf{A}(\boldsymbol{p}) \, V\big)(2I)$ *by* `TempY`*.*

Technical preparations for derivatives of multiple scattering matrix We recall the definitions of $\mathbf{T}$,

$$\begin{aligned}
\mathbf{T}_{II} \quad &= \quad \mathbf{0}_{(2\mathbf{m}+1)\times(2\mathbf{m}+1)} \quad ; \\
(\mathbf{T}_{IJ})_{lm} \quad &= \quad \mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \; e^{i(m-l)\theta_{JI}} = (-1)^{m-l}\mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \; e^{i(m-l)\theta_{IJ}} \quad ; \\
(\mathbf{T}_{JI})_{ml} \quad &= \quad \mathrm{H}^{(1)}_{l-m}\,(\kappa\,\mathsf{d}_{IJ}) \; e^{i(l-m)\theta_{IJ}} = (-1)^{l-m}\mathrm{H}^{(1)}_{l-m}\,(\kappa\,\mathsf{d}_{IJ}) \; e^{i(l-m)\theta_{JI}} \\
&= \quad \mathrm{H}^{(1)}_{m-l}\,(\kappa\,\mathsf{d}_{IJ}) \; e^{i(l-m)\theta_{JI}} \quad .
\end{aligned}$$

with $\mathbf{m}$ denoting the order of the method $1 \leq I, J \leq N_{\text{Obs}}$; $\mathbf{m} \leq l, m \leq \mathbf{m}$. We have also used the identity

$$e^{im\theta_{IJ}} = (-1)^m e^{im\theta_{JI}} \quad .$$

From (73) in C, substitute R_k for $\mathbf{x}_J$, with notation

$$\theta_{IJ} = \theta_I(\mathbf{x}_J) \quad ; \quad \mathsf{d}_{IJ} = \|\mathbf{x}_I - \mathbf{x}_J\|,$$

we have

$$\nabla_{\mathbf{x}_I} e^{\mathrm{i}\, m\, \theta_{IJ}} = \frac{\mathrm{i}\, m\, e^{\mathrm{i}\, m\, \theta_{IJ}}}{\mathsf{d}_{IJ}} \begin{pmatrix} \sin\theta_{IJ} \\ -\cos\theta_{IJ} \end{pmatrix} \quad ; \quad \nabla_{\mathbf{x}_I} \mathsf{d}_{IJ} = \begin{pmatrix} -\cos\theta_{IJ} \\ -\sin\theta_{IJ} \end{pmatrix} \quad .$$

Using the second expression for $(\mathbf{T}_{IJ})_{lm}$, we obtain

$$\begin{aligned} \nabla_{\mathbf{x}_I} (\mathbf{T}_{IJ})_{lm} \;&=\; (-1)^{m-l} \kappa\, \mathrm{H}^{(1)\prime}_{m-l}\, (\kappa\, \mathsf{d}_{IJ})\; e^{i(m-l)\theta_{IJ}} \begin{pmatrix} -\cos\theta_{IJ} \\ -\sin\theta_{IJ} \end{pmatrix} \\ &\quad+\; (-1)^{m-l} \mathrm{H}^{(1)}_{m-l}\, (\kappa\, \mathsf{d}_{IJ})\; \frac{\mathrm{i}\,(m-l)\, e^{\mathrm{i}\,(m-l)\,\theta_{IJ}}}{\mathsf{d}_{IJ}} \begin{pmatrix} \sin\theta_{IJ} \\ -\cos\theta_{IJ} \end{pmatrix} \\ &= e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \left[\kappa\, \mathrm{H}^{(1)\prime}_{m-l}\, (\kappa\, \mathsf{d}_{IJ}) \begin{pmatrix} -\cos\theta_{IJ} \\ -\sin\theta_{IJ} \end{pmatrix} + \frac{\mathrm{i}\,(m-l)\, \mathrm{H}^{(1)}_{m-l}\, (\kappa\, \mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} \sin\theta_{IJ} \\ -\cos\theta_{IJ} \end{pmatrix} \right] \\ &= e^{\mathrm{i}\,(m-l)\,\theta_{JI}} \left[\kappa\, \mathrm{H}^{(1)\prime}_{m-l}\, (\kappa\, \mathsf{d}_{IJ}) \begin{pmatrix} \cos\theta_{JI} \\ \sin\theta_{JI} \end{pmatrix} + \frac{\mathrm{i}\,(m-l)\, \mathrm{H}^{(1)}_{m-l}\, (\kappa\, \mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} -\sin\theta_{JI} \\ \cos\theta_{JI} \end{pmatrix} \right] \quad . \end{aligned} \tag{63}$$

Using the first expression for $(\mathbf{T}_{JI})_{ml}$, we obtain

$$\begin{aligned} &\nabla_{\mathbf{x}_I} (\mathbf{T}_{JI})_{ml} \\ &= e^{\mathrm{i}\,(l-m)\,\theta_{IJ}} \left[\kappa\, \mathrm{H}^{(1)\prime}_{l-m}\, (\kappa\, \mathsf{d}_{IJ}) \begin{pmatrix} -\cos\theta_{IJ} \\ -\sin\theta_{IJ} \end{pmatrix} + \frac{\mathrm{i}\,(l-m)\, \mathrm{H}^{(1)}_{l-m}\, (\kappa\, \mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} \sin\theta_{IJ} \\ -\cos\theta_{IJ} \end{pmatrix} \right] \\ &= (-1)^{m-l} e^{\mathrm{i}\,(l-m)\,\theta_{JI}} \left[\kappa\, \mathrm{H}^{(1)\prime}_{l-m}\, (\kappa\, \mathsf{d}_{IJ}) \begin{pmatrix} -\cos\theta_{IJ} \\ -\sin\theta_{IJ} \end{pmatrix} + \frac{\mathrm{i}\,(l-m)\, \mathrm{H}^{(1)}_{l-m}\, (\kappa\, \mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} \sin\theta_{IJ} \\ -\cos\theta_{IJ} \end{pmatrix} \right] \\ &= e^{\mathrm{i}\,(l-m)\,\theta_{JI}} \left[\kappa\, \mathrm{H}^{(1)\prime}_{m-l}\, (\kappa\, \mathsf{d}_{IJ}) \begin{pmatrix} \cos\theta_{JI} \\ \sin\theta_{JI} \end{pmatrix} + \frac{\mathrm{i}\,(l-m)\, \mathrm{H}^{(1)}_{m-l}\, (\kappa\, \mathsf{d}_{IJ})}{\mathsf{d}_{IJ}} \begin{pmatrix} -\sin\theta_{JI} \\ \cos\theta_{JI} \end{pmatrix} \right] . \end{aligned} \tag{64}$$

B.2 Derivatives of single-layer potential

Since we take the evaluation at receivers and then take the derivative with respect to p, we can just focus on

$$(\mathcal{S}_I\, \mathbf{w}_m)(\mathsf{R}_k) = (\mathcal{S}_I\, \mathbf{w}_m)(r_I(\mathsf{R}_k)\,,\, \theta_I(\mathsf{R}_k)) \quad = \quad \frac{\mathrm{i}\,\pi\, \mathsf{r}_I}{2}\, \mathrm{J}_m(\kappa\, \mathsf{r}_I)\; \mathrm{H}^{(1)}_m\big(\kappa\, r_I(\mathsf{R}_k)\big)\; e^{\mathrm{i}\; m\, \theta_I(\mathsf{R}_k)} \quad . \tag{65}$$

Here r_I and θ_I are the polar coordinates relative to $\mathbf{x}_I$. In particular, for receiver R_k,

$$r_I(\mathsf{R}_k) = \|\mathsf{R}_k - \mathbf{x}_I\| = \Big(\big(\mathsf{R}_{k,1} - \mathbf{x}_{1,1}\big)^2 + \big(\mathsf{R}_{k,2} - \mathbf{x}_{1,2}\big)^2 \Big)^{1/2} \quad ;$$

$$\big(\cos\, \theta_I(\mathsf{R}_k)\,,\, \sin\, \theta_I(\mathsf{R}_k)\big) = \frac{\mathsf{R}_k - \mathbf{x}_I}{\|\mathsf{R}_k - \mathbf{x}_I\|} \quad .$$

From C, c.f. (73), we have calculated

$$\nabla_{\mathbf{x}_I} e^{\mathrm{i}\; m\, \theta_I(\mathsf{R}_k)} = \frac{\mathrm{i}\; m\; e^{\mathrm{i}\; m\, \theta_I(\mathsf{R}_k)}}{\|\mathsf{R}_k - \mathbf{x}_I\|} \begin{pmatrix} \sin\theta_I(\mathsf{R}_k) \\ -\cos\theta_I(\mathsf{R}_k) \end{pmatrix} \quad ; \quad \nabla_{\mathbf{x}_I} r_I(\mathsf{R}_k) \;=\; \begin{pmatrix} -\cos\theta_I(\mathsf{R}_k) \\ -\sin\theta_I(\mathsf{R}_k) \end{pmatrix} \quad .$$

As a result,

$$
\begin{aligned}
&\nabla_{\mathbf{x}_I}\ \mathcal{S}_I\,\mathbf{w}_m(\mathsf{R}_k)\\
&= \quad \frac{\mathrm{i}\,\pi\,\mathsf{r}_I}{2}\ \mathrm{J}_m(\kappa\,\mathsf{r}_I)\ \nabla_{\mathbf{x}_I}\mathrm{H}^{(1)}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)\ e^{\mathrm{i}\ m\,\theta_I(\mathsf{R}_k)} + \frac{\mathrm{i}\,\pi\,\mathsf{r}_I}{2}\ \mathrm{J}_m(\kappa\,\mathsf{r}_I)\ \mathrm{H}^{(1)}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)\ \nabla_{\mathbf{x}_I}e^{\mathrm{i}\ m\,\theta_I(\mathsf{R}_k)}\\
&= \frac{\mathrm{i}\,\pi\,\mathsf{r}_I}{2}\Big[\mathrm{J}_m(\kappa\,\mathsf{r}_I)\ \kappa\,\mathrm{H}^{(1)\prime}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)\ e^{\mathrm{i}\ m\,\theta_I(\mathsf{R}_k)}\begin{pmatrix}-\cos\theta_I(\mathsf{R}_k)\\-\sin\theta_I(\mathsf{R}_k)\end{pmatrix}\\
&\qquad\quad + \mathrm{J}_m(\kappa\,\mathsf{r}_I)\ \mathrm{H}^{(1)}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)\,\frac{\mathrm{i}\ m\ e^{\mathrm{i}\ m\,\theta_I(\mathsf{R}_k)}}{\|\mathsf{R}_k-\mathbf{x}_I\|}\begin{pmatrix}\sin\theta_I(\mathsf{R}_k)\\-\cos\theta_I(\mathsf{R}_k)\end{pmatrix}\Big]\\
&= \frac{\mathrm{i}\,\pi\,\mathsf{r}_I}{2}\mathrm{J}_m(\kappa\,\mathsf{r}_I)e^{\mathrm{i}\ m\,\theta_I(\mathsf{R}_k)}\Big[-\kappa_e\,\mathrm{H}^{(1)\prime}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)\ \begin{pmatrix}\cos\theta_I(\mathsf{R}_k)\\\sin\theta_I(\mathsf{R}_k)\end{pmatrix}\\
&\qquad\quad + \ \frac{\mathrm{i}\ m\,\mathrm{H}^{(1)}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)}{\|\mathsf{R}_k-\mathbf{x}_I\|}\begin{pmatrix}\sin\theta_I(\mathsf{R}_k)\\-\cos\theta_I(\mathsf{R}_k)\end{pmatrix}\Big]\,.
\end{aligned}
$$

The matrix $\partial_p\Big(\mathfrak{R}(\boldsymbol{p})\,V\Big)$ with V independent of $\boldsymbol{p}$ is of size $N_{\text{rec}}\times N_{\text{par}}$. Its component at k-th row and between $2I$-th and $2I+1$ column is the row vector of length 2,

$$
\boxed{
\begin{aligned}
&\Big[\partial_p\Big(\mathfrak{R}(\boldsymbol{p})\,V\Big)\Big](k\,,\,2I-1:1I)\\
&=\partial_p\sum_{1\leq J\leq N_{\text{Obs}}}\ \sum_{-\mathbf{m}\leq m\leq\mathbf{m}}V_{J,l}\ (\mathcal{S}_J\mathbf{w}_l)(\mathsf{R}_k)\quad=\quad\sum_{1\leq J\leq N_{\text{Obs}}}\ \sum_{-\mathbf{m}\leq m\leq\mathbf{m}}V_{J,l}\ \partial_p\,(\mathcal{S}_J\mathbf{w}_l)(\mathsf{R}_k)\\
&=\sum_{-\mathbf{m}\leq m\leq\mathbf{m}}\frac{\mathrm{i}\,\pi\,\mathsf{r}_I}{2}\ \mathrm{J}_m(\kappa\,\mathsf{r}_I)e^{\mathrm{i}\ m\,\theta_I(\mathsf{R}_k)}\ V_{Im}\Big[-\kappa_e\,\mathrm{H}^{(1)\prime}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)\begin{pmatrix}\cos\theta_I(\mathsf{R}_k)\\\sin\theta_I(\mathsf{R}_k)\end{pmatrix}^t\\
&\qquad\quad+\ \frac{\mathrm{i}\ m\,\mathrm{H}^{(1)}_m\big(\kappa\,r_I(\mathsf{R}_k)\big)}{\|\mathsf{R}_k-\mathbf{x}_I\|}\begin{pmatrix}\sin\theta_I(\mathsf{R}_k)\\-\cos\theta_I(\mathsf{R}_k)\end{pmatrix}^t\Big].
\end{aligned}
}
\tag{66}
$$

Remark 16. *In the code, the matrix* $\partial_{\boldsymbol{p}}\mathfrak{R}\underset{(J,l)}{\cdot}\mathsf{S}(\boldsymbol{p})$ *is calculated by routine*

Grad_ SLMatrix *(BesselJ_ RadObs , NObs , NRec , PolarCoMat , radiusObs , wavenumber OrdApp , Exp_ RecMat , DerHankel_ Rec , Hankel_ Rec , sol , sizeSol , Grad_ SLMat)*

B.3 Derivatives of the source term associated with planewave

For the incident planewave defined as

$$u_{\text{pw}}(\mathbf{x}) \quad = \quad e^{\mathrm{i}\,\kappa\,\mathbf{x}\cdot(\cos\alpha_{\text{inc}},\sin\alpha_{\text{inc}})}\,,$$

the RHS of (10) is given by

$$F(\boldsymbol{p},u_{\text{pw}}) \ = \quad F(\boldsymbol{p},u_{\text{pw}}) \ = \ \mathbf{D}_3\ F_{\text{pw}} \quad ,$$

where F_{pw} is a vector of size $N=(2\mathbf{m}+1)N_{\text{Obs}}$ with components,

$$(F_{\text{pw}})_{Il} \quad = \quad -2\,u_{\text{pw}}(\mathbf{x}_I)\ \mathrm{i}^l\ e^{-\mathrm{i}\,l\,\alpha_{\text{inc}}} \quad , \quad 1\leq I\leq N_{\text{Obs}}\,,\,-\mathbf{m}\leq l\leq\mathbf{m} \quad .$$

Since $\mathbf{D}_3$ is independent of the position of the obstacles, we have

$$\partial_{\boldsymbol{p}}\, F(\boldsymbol{p}, u_{\mathrm{pw}}) = \mathbf{D}_3 \; \partial_{\boldsymbol{p}}\, F_{\mathrm{pw}}\,.$$

We next calculate the components of matrix $\partial_{\boldsymbol{p}} F_{\mathrm{pw}}$, the latter is of size $N \times N_{\mathrm{par}}$.

$$\partial_{\mathbf{x}_J} u_{\mathrm{pw}}(\mathbf{x}_I) \quad = \quad \mathrm{i}\,\kappa\, u_{\mathrm{pw}}(\mathbf{x}_I) \begin{pmatrix} \cos\alpha_{\mathrm{inc}} \\ \sin\alpha_{\mathrm{inc}} \end{pmatrix}^t \delta_{IJ}$$

we have

$$\begin{aligned} \partial_{\mathbf{x}_I}\, (F_{\mathrm{pw}})_{Il} \quad &= \quad -2\,\mathrm{i}^{l+1}\; e^{-\mathrm{i}\, l\alpha_{\mathrm{inc}}}\, \kappa\, u_{\mathrm{pw}}(\mathbf{x}_I) \begin{pmatrix} \cos\alpha_{\mathrm{inc}} \\ \sin\alpha_{\mathrm{inc}} \end{pmatrix}^t \quad ; \\ \partial_{\mathbf{x}_J}\, (F_{\mathrm{pw}})_{Il} \quad &= \quad \mathbf{0}_{1\times 2} \quad , \quad I \neq J\,. \end{aligned} \tag{67}$$

These are also the components at row (I,l) and columns between $2J-1:2J$ of matrix $\partial_{\boldsymbol{p}} F_{\mathrm{pw}}$.

For row vector W of length N, the quantity $W^t \partial_{\boldsymbol{p}}\; F(\boldsymbol{p}, \mathbf{u}_{\mathrm{inc}})$ is a column vector of length $N_{\mathrm{par}} = 2N_{\mathrm{Obs}}$, with components at levels $2I-1:2I$ given by

$$\begin{aligned} &\big(\, W^t\; \partial_{\boldsymbol{p}}\, F(\boldsymbol{p}, u_{\mathrm{pw}})\,\big)(2I-1:2I) \quad = \quad W^t\; \mathbf{D}_3\; \partial_{\mathbf{x}_I} F_{\mathrm{pw}} \\ &= \sum_{J=1}^{N_{\mathrm{Obs}}} \sum_{l=-\mathbf{m}}^{\mathbf{m}} W_{Jl}\; (\partial_{\mathbf{x}_I} F_{\mathrm{pw}})(Jl,:)\; (\mathbf{D}_3)_{Jl} \\ &= -2\,\kappa \sum_{l=-\mathbf{m}}^{\mathbf{m}} W_{Il}\; \mathrm{i}^{l+1}\; e^{-il\alpha_{\mathrm{inc}}}\; u_{\mathrm{pw}}(\mathbf{x}_I) \begin{pmatrix} \cos\alpha_{\mathrm{inc}} \\ \sin\alpha_{\mathrm{inc}} \end{pmatrix}^t \times \begin{cases} \mathrm{J}_l(\kappa\, \mathsf{r}_I) & ,\text{Dirichlet} \\ \kappa\; \mathrm{J}_l'(\kappa\, \mathsf{r}_I) & ,\text{Neumann} \end{cases} . \end{aligned}$$

After some rearrangement, we arrive at: for row vector W of length N

$$\begin{aligned} &\big(W^t\; \partial_{\boldsymbol{p}}\, F(\boldsymbol{p}, u_{\mathrm{pw}})\big)(2I-1:2I) \\ &= -2\,\kappa\, u_{\mathrm{pw}}(\mathbf{x}_I) \begin{pmatrix} \cos\alpha_{\mathrm{inc}} \\ \sin\alpha_{\mathrm{inc}} \end{pmatrix}^t \sum_{l=-\mathbf{m}}^{\mathbf{m}} W_{Il}\; \mathrm{i}^{l+1}\; e^{-\mathrm{i}\, l\, \alpha_{\mathrm{inc}}} \times \begin{cases} \mathrm{J}_l(\kappa\, \mathsf{r}_I) & ,\text{Dirichlet} \\ \kappa\; \mathrm{J}_l'(\kappa\, \mathsf{r}_I) & ,\text{Neumann} \end{cases} \end{aligned} \tag{68}$$

C Derivatives of Polar coordinates

We write

$$\mathsf{R}_k = (\mathsf{R}_{k,1}, \mathsf{R}_{k,2}) \quad ; \quad \mathbf{x}_I = (\mathbf{x}_{I,1}, \mathbf{x}_{I,2}) \quad .$$

We recall

$$r_J(\mathsf{R}_k) = \|\mathsf{R}_k - \mathbf{x}_J\| = \Big(\big(\mathsf{R}_{k,1} - \mathbf{x}_{J,1}\big)^2 + \big(\mathsf{R}_{k,2} - \mathbf{x}_{J,2}\big)^2 \Big)^{1/2} \quad .$$

$$\cos\theta_J(\mathsf{R}_k) = \frac{\mathsf{R}_{k,1} - \mathbf{x}_{J,1}}{\|\mathsf{R}_k - \mathbf{x}_J\|} \quad ; \quad \sin\theta_J(\mathsf{R}_k) = \frac{\mathsf{R}_{k,2} - \mathbf{x}_{J,2}}{\|\mathsf{R}_k - \mathbf{x}_J\|} \quad .$$

Also note that

$$\big(\mathsf{R}_{k,1} - \mathbf{x}_{J,1}\big) \,+\, \mathrm{i}\,\big(\mathsf{R}_{k,2} - \mathbf{x}_{J,2}\big) \quad = \quad \|\mathsf{R}_k - \mathbf{x}_J\|\, e^{\mathrm{i}\;\theta_J(\mathsf{R}_k)} \quad . \tag{69}$$

We assume $\mathsf{R}_k \neq \mathbf{x}_I$. For $J \neq I$,

$$\partial_{\mathbf{x}_{I,i}} r_J(\mathsf{R}_k) = \partial_{\mathbf{x}_{I,i}} \theta_J(\mathsf{R}_k) = 0 \quad , \quad \forall\, 1 \leq k \leq N_{\mathbf{rec}} \; ; \; i = 1,2 \quad .$$

It remains to calculate for $J = I$. We first consider the polar radius

$$\partial_{\mathbf{x}_{I,1}} r_I(\mathsf{R}_k) = -\frac{(\mathsf{R}_{k,1} - \mathbf{x}_{1,1})}{\|\mathsf{R}_k - \mathbf{x}_I\|} = -\cos\theta_I(\mathsf{R}_k) \;\; ; \partial_{\mathbf{x}_{I,2}} r_I(\mathsf{R}_k) \quad = -\frac{(\mathsf{R}_{k,2} - \mathbf{x}_{1,2})}{\|\mathsf{R}_k - \mathbf{x}_I\|} = -\sin\theta_I(\mathsf{R}_k)\,. \tag{70}$$

Next, we consider the polar angles,

$$\cos\,\theta_I(\mathsf{R}_k) = \frac{\mathsf{R}_{k,1} - \mathbf{x}_{I,1}}{\|\mathsf{R}_k - \mathbf{x}_I\|} \quad \Longrightarrow \quad \begin{cases} -\sin\theta_I(\mathsf{R}_k)\,\partial_{\mathbf{x}_{1,1}}\theta_I(\mathsf{R}_k) \;=\; -\dfrac{1}{\|\mathsf{R}_k - \mathbf{x}_I\|} + \dfrac{(\mathsf{R}_{k,1} - \mathbf{x}_{I,1})^2}{\|\mathsf{R}_k - \mathbf{x}_I\|^3} & ; \\ -\sin\theta_I(\mathsf{R}_k)\,\partial_{\mathbf{x}_{1,2}}\theta_I(\mathsf{R}_k) \;=\; \dfrac{(\mathsf{R}_{k,1} - \mathbf{x}_{I,1})(\mathsf{R}_{k,2} - \mathbf{x}_{I,2})}{\|\mathsf{R}_k - \mathbf{x}_I\|^3} & \end{cases} \quad .$$

$$\sin\,\theta_I(\mathsf{R}_k) = \frac{\mathsf{R}_{k,2} - \mathbf{x}_{I,2}}{\|\mathsf{R}_k - \mathbf{x}_I\|} \quad \Longrightarrow \quad \begin{cases} \cos\,\theta_I(\mathsf{R}_k)\,\partial_{\mathbf{x}_{1,1}}\theta_I(\mathsf{R}_k) = \dfrac{(\mathsf{R}_{k,2} - \mathbf{x}_{I,2})(\mathsf{R}_{k,1} - \mathbf{x}_{I,1})}{\|\mathsf{R}_k - \mathbf{x}_I\|^3} & ; \\ \cos\,\theta_I(\mathsf{R}_k)\,\partial_{\mathbf{x}_{1,2}}\theta_I(\mathsf{R}_k) = -\dfrac{1}{\|\mathsf{R}_k - \mathbf{x}_I\|} + \dfrac{(\mathsf{R}_{k,2} - \mathbf{x}_{I,2})^2}{\|\mathsf{R}_k - \mathbf{x}_I\|^3} & \end{cases} \quad .$$

Using the above expressions, we calculate

$$\begin{aligned} e^{\mathrm{i}\,\theta_I(\mathsf{R}_k)}\partial_{\mathbf{x}_{1,1}}\theta_I(\mathsf{R}_k) &= \Big(\cos\theta_I(\mathsf{R}_k) + \mathrm{i}\,\sin\theta_I(\mathsf{R}_k)\Big)\,\partial_{\mathbf{x}_{1,1}}\theta_I(\mathsf{R}_k) \\ &= \frac{1}{\|\mathsf{R}_k - \mathbf{x}_I\|^3}\Big((\mathsf{R}_{k,2} - \mathbf{x}_{I,2})(\mathsf{R}_{k,1} - \mathbf{x}_{I,1}) - \mathrm{i}(\mathsf{R}_{k,1} - \mathbf{x}_{I,1})^2 + \mathrm{i}\|\mathsf{R}_k - \mathbf{x}_I\|^2\Big) \\ &= \frac{1}{\|\mathsf{R}_k - \mathbf{x}_I\|^3}\Big(-\mathrm{i}(\mathsf{R}_{k,1} - \mathbf{x}_{I,1})\,e^{\mathrm{i}\,\theta_I}\|\mathsf{R}_k - \mathbf{x}_I\| + \mathrm{i}\|\mathsf{R}_k - \mathbf{x}_I\|^2\Big) \quad . \end{aligned}$$

In the second equality, we have used (69). After dividing by $e^{\mathrm{i}\,\theta_I(\mathsf{R}_k)}$ on both sides, we obtain

$$\begin{aligned} \partial_{\mathbf{x}_{1,1}}\theta_I(\mathsf{R}_k) \;&=\; \frac{-\mathrm{i}(\mathsf{R}_{k,1} - \mathbf{x}_{I,1})}{\|\mathsf{R}_k - \mathbf{x}_I\|^2} + \frac{\mathrm{i}\,e^{-\mathrm{i}\,\theta_I(\mathsf{R}_k)}}{\|\mathsf{R}_k - \mathbf{x}_I\|} \\ &=\; \frac{-\mathrm{i}\cos\theta_I(\mathsf{R}_k) + \mathrm{i}\big(\cos\theta_I(\mathsf{R}_k) - \mathrm{i}\sin\theta_I(\mathsf{R}_k)\big)}{\|\mathsf{R}_k - \mathbf{x}_I\|} \;=\; \frac{\sin\theta_I(\mathsf{R}_k)}{\|\mathsf{R}_k - \mathbf{x}_I\|}\,. \end{aligned} \tag{71}$$

Similarly, we obtain

$$\begin{aligned} e^{\mathrm{i}\,\theta_I(\mathsf{R}_k)}\partial_{\mathbf{x}_{1,2}}\theta_I(\mathsf{R}_k) &= \Big(\cos\theta_I(\mathsf{R}_k) + \mathrm{i}\,\sin\theta_I(\mathsf{R}_k)\Big)\,\partial_{\mathbf{x}_{1,2}}\theta_I(\mathsf{R}_k) \\ &= \frac{1}{\|\mathsf{R}_k - \mathbf{x}_I\|^3}\Big(-\mathrm{i}\,(\mathsf{R}_{k,2} - \mathbf{x}_{I,2})(\mathsf{R}_{k,1} - \mathbf{x}_{I,1}) + (\mathsf{R}_{k,2} - \mathbf{x}_{I,2})^2 - \|\mathsf{R}_k - \mathbf{x}_I\|^2\Big) \\ &= \frac{1}{\|\mathsf{R}_k - \mathbf{x}_I\|^3}\Big(-\mathrm{i}(\mathsf{R}_{k,2} - \mathbf{x}_{I,2})\,e^{\mathrm{i}\,\theta_I}\,\|\mathsf{R}_k - \mathbf{x}_I\| - \|\mathsf{R}_k - \mathbf{x}_I\|^2\Big) \quad . \end{aligned}$$

After dividing by $e^{\mathrm{i}\,\theta_I(\mathsf{R}_k)}$ on both sides, we obtain

$$\begin{aligned} \partial_{\mathbf{x}_{1,2}}\theta_I(\mathsf{R}_k) &= \frac{-\mathrm{i}(\mathsf{R}_{k,2} - \mathbf{x}_{I,2})}{\|\mathsf{R}_k - \mathbf{x}_I\|^2} - \frac{e^{-\mathrm{i}\theta_I(\mathsf{R}_k)}}{\|\mathsf{R}_k - \mathbf{x}_I\|} \\ &= \frac{-\mathrm{i}\sin\theta_I(\mathsf{R}_k) - \Big(\cos\theta_I(\mathsf{R}_k) - \mathrm{i}\sin\theta_I(\mathsf{R}_k)\Big)}{\|\mathsf{R}_k - \mathbf{x}_I\|} = \frac{-\cos\theta_I(\mathsf{R}_k)}{\|\mathsf{R}_k - \mathbf{x}_I\|}\,. \end{aligned} \tag{72}$$

As a result

$$\boxed{\begin{array}{rcccl} \nabla_{\mathbf{x}_I} e^{\mathrm{i}\,\theta_I(\mathsf{R}_k)} & = & \mathrm{i}\,e^{\mathrm{i}\,\theta_I(\mathsf{R}_k)} \begin{pmatrix}\partial_{\mathbf{x}_{I,1}}\theta_I(\mathsf{R}_k)\\ \partial_{\mathbf{x}_{I,2}}\theta_I(\mathsf{R}_k)\end{pmatrix} & = & \dfrac{\mathrm{i}\,e^{\,\theta_I(\mathsf{R}_k)}}{\|\mathsf{R}_k - \mathbf{x}_I\|}\begin{pmatrix}\sin\theta_I(\mathsf{R}_k)\\ -\cos\theta_I(\mathsf{R}_k)\end{pmatrix} \\ \nabla_{\mathbf{x}_I} r_I(\mathsf{R}_k) & = & -\dfrac{\mathsf{R}_k - \mathbf{x}_I}{\|\mathsf{R}_k - \mathbf{x}_I\|} & = & \begin{pmatrix}-\cos\theta_I(\mathsf{R}_k)\\ -\sin\theta_I(\mathsf{R}_k)\end{pmatrix} \quad . \end{array}} \tag{73}$$

D More on Wolfe Linesearch Algorithm

Recall that we have defined

$$\phi(\alpha) \;:= \widehat{\mathcal{J}}(\boldsymbol{p} + \alpha\mathbf{s}).$$

Assumption $\phi'(0) < 0$ ($\mathbf{p}$ is a descent direction). Define

$$\mathbf{T}_{\mu,\eta} := \left\{ \alpha > 0 \;\middle|\; \begin{array}{c} \phi(\alpha) \;\leq\; \phi(0) + \alpha\,\mu\,\phi'(0) \\ \\ |\phi'(\alpha)| \;\leq\; \eta\,|\phi'(0)| \end{array} \right\} \quad , \quad \mathbf{T}_{\mu} := \mathbf{T}_{\mu,\mu}\,.$$

In the Wolfe line search algorithm, we look for α that satisfies the strong Wolfe conditions (44), which is equivalent to requiring $\alpha \in \mathbf{T}_{\mu,\eta}$.

We use the following two theorems from [39].

Theorem 5 (Theorem 2.1 in [39])**.** *Define the auxiliary function by*

$$\psi(\alpha) := \phi(\alpha) - \phi(0) - \mu\,\alpha\,\phi'(0)\,. \tag{74}$$

Assume $\alpha_l \neq \alpha_u$ but not ordered, and let I be a closed interval with endpoints α_l and α_u. If the endpoints satisfy

$$\psi(\alpha_l) \leq \psi(\alpha_u)\,, \quad \psi(\alpha_l) \leq 0\,, \quad \psi'(\alpha_l)(\alpha_u - \alpha_l) < 0 \tag{75}$$

then there is an $\alpha_\star$ in I with

$$\psi(\alpha_\star) \leq \psi(\alpha_l) \quad , \quad \psi'(\alpha_\star) = 0\,.$$

In particular, $\alpha_\star \in T(\mu) \cap I$.

Theorem 6 (Theorem 3.2 in [39])**.** *Consider closed interval I with endpoints α_l and α_u (not ordered). If the endpoints satisfy*

$$\phi(\alpha_l) \leq \phi(\alpha_u) \qquad \text{and} \qquad \phi'(\alpha_l)(\alpha_u - \alpha_l) < 0$$

then there is an $\alpha_\star \in I$ with $\phi(\alpha_\star) \leq \phi(\alpha_l)$ and $\phi'(\alpha_\star) = 0$.

Remark 17. *The first assumption gives that α_l is the point with lower value of ϕ. The meaning of the second assumption is that $\overrightarrow{(\alpha_l, \alpha_u)}$ is a descent direction for ϕ at α_l. Thus ϕ can be decreased by searching near α_l in this direction.*

Proposition 7. *Assumption $\eta \geq \mu$. Consider an interval $(\alpha_{\min}, \alpha_{\max})$ with $\alpha_{\min}$ satisfying:*

$$\phi'(\alpha_{\min})(\alpha_{\max} - \alpha_{\min}) < 0. \tag{76}$$

and

$$\alpha_{\min} \textit{ satisfies the sufficient decrease condition but fails the curvature condition.} \tag{77}$$

This interval $(\alpha_{\min}, \alpha_{\max})$ contains step lengths satisfying the strong Wolfe condition if one of the conditions is satisfied:

(i) $\alpha_{\max}$ *violates the sufficient decrease condition (and* $\phi'(\alpha_{\max}) < 0$*),*

(ii) $\phi(\alpha_{\max}) > \phi(\alpha_{\min})$*,*

(iii) $\phi'(\alpha_{\max}) \geq 0$*.*

Proof. Define

$$\psi(\alpha) := \phi(\alpha) - \mu\,\alpha\,\phi'(0) \quad \Rightarrow \quad \psi'(\alpha) = \phi'(\alpha) - \mu\,\phi'(0)\,.$$

The current assumption is that $\alpha_{\min}$ satisfies the sufficient decrease condition, i.e.

$$\phi(\alpha) \;<\; \mu\,\alpha\,\phi'(0) \quad \Rightarrow \quad \psi(\alpha_{\min}) > 0\,,$$

but fails the curvature condition i.e.

$$\phi'(\alpha_{\min}) \;<\; \eta\,\phi'(0) \quad \text{or} \quad -\,\eta\phi'(0) \;<\; \phi'(\alpha_{\min}). \tag{78}$$

- If $\alpha_{\max} > \alpha_{\min}$, then assumption (76) implies $\phi'(\alpha_{\min}) < 0$, and inequalities (78) thus reduces to $\phi'(\alpha_{\min}) \;<\; \eta\,\phi'(0)$. Since $\eta \geq \mu > 0$ and $\phi'(0) < 0$,

  $$\phi'(\alpha_{\min}) \;<\; \eta\,\phi'(0) \;\leq\; \mu\phi'(0) \quad \Rightarrow \quad \psi'(\alpha_{\min}) \;<\; 0\,.$$

- If $\alpha_{\max} < \alpha_{\min}$, then assumption (76) implies $\phi'(\alpha_{\min}) > 0$, and inequalities (78) thus reduce to $-\eta\,\phi'(0) \;<\; \phi'(\alpha_{\min})$. With $\eta \geq \mu > 0$ and $\phi'(0) < 0$

  $$-\mu\phi'(0) \;<\; -\eta\phi'(0) \;<\; \phi'(\alpha_{\min}) \quad \Rightarrow \quad \phi'(\alpha_{\min}) + \mu\,\phi'(0) > 0$$

  At the same time, since $\phi'(0) < 0$,

  $$\phi'(\alpha_{\min}) - \mu\,\phi'(0) \;>\; \phi'(\alpha_{\min}) + \mu\,\phi'(0)\,.$$

  This means

  $$\psi'(\alpha_{\min}) \;=\; \phi'(\alpha_{\min}) - \eta\,\phi'(0) > 0.$$

As a result, in both cases, we have

$$\psi'(\alpha_{\min})\,(\alpha_{\max} - \alpha_{\min}) \;<\; 0\,. \tag{79}$$

Condition (i): If $\alpha_{\max}$ violates the sufficient decrease condition, $\psi(\alpha_{\max}) > 0$. The endpoints $\alpha_{\min}$ and $\alpha_{\max}$ satisfy

$$\psi(\alpha_{\min}) \;<\; 0 \;<\; \psi(\alpha_{\max}) \quad , \quad \psi'(\alpha_{\min})\,(\alpha_{\max} - \alpha_{\min}) < 0\,.$$

This together with (79) means that the interval with endpoints $\alpha_{\min}$ and $\alpha_{\max}$ satisfies the condition of Theorem 2.1 in [39] with $\alpha_l = \alpha_{\min}$ and $\alpha_u = \alpha_{\max}$. Hence there exits $\alpha_\star \in (\alpha_{\min}, \alpha_{\max})$ that satisfies the strong Wolfe condition.

Condition (ii) : Can assume here β does not violate the sufficient decrease condition otherwise we use the proof for condition (i). Use Theorem 3.2 in [39] to give that the global minimum $\alpha_\star$ in the interior of $(\alpha_{\min}, \alpha_{\max})$ such that $\phi(\alpha_\star) \leq \phi(\alpha_{\min})$ and $\phi'(\alpha_\star) = 0$. For any $x \in (\alpha, \beta)$ such that $\phi(x) < \phi(\beta)$ we have

$$\phi(x) \;<\; \phi(\beta) \;<\; \phi(0) + \mu\beta\phi'(0) \;<\; \phi + \mu x\phi'(0)$$

thus the global minimum on (α, β) satisfies $\phi'(\alpha_\star) = 0$ and the sufficient decrease condition.

Condition (iii) : Can assume here β does not violate the sufficient decrease condition otherwise we use the proof for condition (i). If $\phi'(\alpha_{\max}) = 0$ then $\alpha_{\max}$ is the point that satisfies the strong Wolfe condition.

If $\phi'(\alpha_{\max}) > 0$, the global minimum is in the interior since $\alpha_{\min} < \alpha_\star$ (due to $\phi'(\alpha_{\min} < 0)$ and $\alpha_\star < \alpha_{\max}$ (due to $\phi'(\alpha_{\max} > 0)$. Next, for any $x \in (\alpha, \beta)$ such that $\phi(x) < \phi(\beta)$ we have

$$\phi(x) \;<\; \phi(\beta) \;<\; \phi(0) \;+\; \mu\,\beta\,\phi'(0) \;<\; \phi \;+\; \mu\,x\,\phi'(0)$$

thus the global minimum on (α, β) satisfies $\phi'(\alpha_\star) = 0$ and the sufficient decrease condition.

□

E Detail of parameters for the experiments

In this appendix we detail the precise set of parameters used for all eight experiments that has been shown.

E.1 Six obstacles reconstruction: noise free – IG2

Common parameters: Order FSSL = 3, $N_{\text{Acq}} = 1$, $\mu = 1.0 \times 10^{-4}$ (for LS2 and LS3), $\eta = 0.9$ (for LS 3).

$$N_{\text{IterMax}} = 60\ ,\ N_{\text{LS IterMax 1}} = 30\quad N_{\text{LS IterMax 2}} = 30\ \text{(for LS 3)}\,.$$

Parameters for SD1 - LS1

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5}\ ,\ \epsilon_{\text{Stag J}} = 0.1\ ,\ \epsilon_{\text{Stag Pos}} = \begin{cases} 5\times10^{-2}\ , \text{run 1-9} \\ 5\times10^{-3}\ , \text{run 10-12} \end{cases}\qquad;$$

Run	κ	**Reference step length**
$1, 2$	$0.08, 0.09$	10
$3, 4$	$0.1, 0.2$	8
5	0.3	6
$6, 7, 8$	$0.4, 0.5, 0.8$	5
9	0.9	4
10	1.0	3
$11, 12$	$1.1, 1.2$	2

.

Parameters for SD1-LS2

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5}\ ;\ \epsilon_{\text{Stag J}} = 0.1\ ,\ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.01\ , \text{run 1} \\ 0.05\ , \text{run 2-9} \\ 5\times10^{-3}\ , \text{run 10-12} \end{cases}\qquad;$$

Run	κ	**Reference step length**
$1, 2$	$0.08, 0.09$	8
$3 - 8$	$0.1 : 0.1 : 0.6$	5
9	0.7	2
$10, 11, 12$	$0.8, 0.9, 1.0$	1

.

Parameters for SD1-LS3

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.01 \text{ , run 1-10} \\ 1 \times 10^{-3} \text{ , run 10-12} \end{cases} \quad ;$$

Run	κ	Reference step length
$1, 2$	$0.08, 0.09$	8
$3, 4$	$0.1, 0.2$	5
$5, 6$	$0, 3, 0.4$	4
$7, 8, 9$	$0.5, 0.6, 0.7$	3
$10, 11, 12$	$0.8, 0.9, 1.0$	1

.

Parameters for SD2-LS1

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-6} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.5 \text{ , run 1-6} \\ 0.05 \text{ , run 7} \\ 1 \times 10^{-3} \text{ , run 8,9} \end{cases} \quad ;$$

Run	κ	Reference step length
$1, 2$	$0.08, 0.09$	10
$3, 4$	$0.1, 0.2$	8
$5, 6, 7$	$0, 3, 0.4, 0.5$	6
$8, 9$	$0.8, 0.9$	5

.

Parameters for SD2-LS2

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-6} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.5 \text{ , run 1-7} \\ 1.0 \times 10^{-3} \text{ , run 8-12} \\ 1 \times 10^{-3} \text{ , run 8,9} \end{cases} \quad ;$$

Run	κ	Reference step length
$1, 2, 3$	$0.08, 0.09, 0.1$	10
$4, 5$	$0.2, 0.3$	8
$6, 7$	$00.4, 0.5$	5
$8 - 12$	$0.6 : 0.1 : 1.0$	2

.

Parameters for SD2-LS3

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = \begin{cases} 10^{-5} \text{ run 1-8} \\ 10^{-6} \text{ run 9-12} \end{cases} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.1 \text{ run 1} \\ 0.5 \text{ , run 2-6} \\ 5.0 \times 10^{-3} \text{ , run 7-12} \end{cases} \quad ;$$

Run	κ	Reference step length
$1, 2$	$0.08, 0.09$	8
$3, 4$	$0.1, 0.2$	5
$5, 6$	$0.3, 0.4$	4
$7, 8, 9$	$0.5, 0.6, 0.7$	3
10	0.8	2
$11, 12$	$0.9, 1.0$	1

.

E.2 Six obstacles reconstruction: noisy data – IG1

Common parameters: Order FSSL = 3, $N_{\text{Acq}} = 3$, $\mu = 1.0 \times 10^{-4}$ (for LS2 and LS3).

$$N_{\text{IterMax}} = 90 \ , \ N_{\text{LS IterMax 1}} = 30 \ \ N_{\text{LS IterMax 2}} = 30 \text{ (for LS 3)} \, .$$

Parameters for SD1-LS1

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.5 \text{ run 1-5} \\ 0.05 \text{ , run 6} \\ 5.0 \times 10^{-3} \text{ , run 7-9} \end{cases}$$

Run	κ	Reference step length
1	0.08	1
2	0.09	8
$3, 4$	$0.1, 0.2$	5
5	0.3	4
6	0.4	2
$7, 8, 9$	$0.5, 0.6, 0.7$	1

.

Parameters for SD1-LS2

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.05 \text{ , run 1- 6} \\ 5.0 \times 10^{-3} \text{ , run 7-8} \end{cases} ;$$

Run	κ	Reference step length
1	0.08	2
$2 - 6$	$0.09, 0.1 : 0.1 : 0.4$	3
$7, 8$	$0.5, 0.6$	2

.

Parameters for SD1-LS3 $\eta = 0.9$

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.05 \text{ , run 1- 7} \\ 5.0 \times 10^{-3} \text{ , run 8-9} \end{cases} ;$$

Run	κ	Reference step length
1	0.08	1
$2, 3$	$0.09, 0, 1$	12
$4 - 9$	$0.2 : 0.1 : 0.7$	6

.

Parameters for SD2-LS1

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.5 \text{ , run 1- 8} \\ 0.05 \text{ , run 9} \end{cases} ;$$

Run	κ	Reference step length
1	0.08	12
2	0.09	8
$3, 4$	$0.1, 0.2$	5
5	0.3	4
6	0.4	2
$7, 8, 9$	$0.5, 0.6, 0.7$	1

.

Parameters for SD2-LS2

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.05 \ , \text{run 1- 3} \\ 5.0 \times 10^{-3} \ , \text{run 4- 9} \end{cases} \ ;$$

Run	κ	Reference step length
1	0.08	6
2	0.09	5
$3, 4, 5$	$0.1, 0.2, 0.3$	3
$6 - 9$	$0.4 : 0.1 : 07$	2

.

Parameters for SD2-LS3 $\eta = 0.4$

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.05 \ , \text{run 1- 7} \\ 5.0 \times 10^{-3} \ , \text{run 8, 9} \end{cases} \ ;$$

Run	κ	Reference step length
1	0.08	1
$2, 3$	$0.09, 0.1$	12
$4 - 9$	$0.2 : 0.1 : 0.7$	6

.

E.3 Six obstacles reconstruction: noisy data – IG2

Common parameters: Order FSSL = 3, $N_{\text{Acq}} = 3$, $\mu = 1.0 \times 10^{-4}$ (for LS2 and LS3).

$$N_{\text{IterMax}} = 60 \ , \ N_{\text{LS IterMax 1}} = 30 \ \ N_{\text{LS IterMax 2}} = 30 \text{ (for LS 3)} \, .$$

Parameters for SD1-LS1

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = 0.05$$

Run	κ	Reference step length
1	0.08	10
$2, 3, 4$	$0.09, 0.1, 0.2$	4
$5 - 9$	$0.3 : 0.1 : 0.7$	3

.

Parameters for SD1-LS2

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.05 \ , \text{run 1- 4} \\ 5.0 \times 10^{-3} \ , \text{run 5-8} \end{cases}$$

Run	κ	Reference step length
1	0.08	6
2	0.09	5
$3, 4, 5$	$0.1, 0.2, 0.3$	4
$6, 7, 8$	$0.4, 0.5, 0.6$	2

.

Parameters for SD1-LS3

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = 5.0 \times 10^{-3}$$

Run	κ	**Reference step length**
1	0.08	1
2	0.09	12
3, 4	0.1, 0.2	4
5 − 8	0.3 : 0.1 : 0.6	2

.

Parameters for SD2-LS1

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 0.05 \text{ , run 1- 4} \\ 5.0 \times 10^{-3} \text{ , run 5-9} \end{cases}$$

Run	κ	**Reference step length**
1, 2	0.08, 0.09	5
3 − 9	0.1 : 0.1 : 0.7	4
10	0.8	3

.

Parameters for SD2-LS2

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = 1.0 \times 10^{-3}$$

Run	κ	**Reference step length**
1	0.08	6
2	0.09	5
3, 4, 5	0.1, 0.2, 0.3	4
6 − 9	0.4 : 0.1 : 0.7	2

.

Parameters for SD2-LS3

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 5.0 \times 10^{-3} \text{ , run 1- 4} \\ 5.0 \times 10^{-4} \text{ , run 5-8} \end{cases}$$

Run	κ	**Reference step length**
1	0.08	1
2	0.09	12
3, 4	0.1, 0.2	4
5 − 8	0.3 : 0.1 : 0.6	2

.

E.4 Twelve obstacles reconstruction: noise free

Parameters for method SD1-LS3

- 10 frequencies with one recycle for a total 11 runs: κ = 0.09,0.1:0.1:0.5, 0.4 (recycled), 0.6, 0.7.
- Common to all runs: $N_{\text{IterMax}} = 70$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.

- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 1.0 \times 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 5.0 \times 10^{-3} & \text{Run 1–5} \\ 5.0 \times 10^{-4} & \text{Run 6,7} \\ 5.0 \times 10^{-5} & \text{Run 8,9} \end{cases}$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.09	3	1
2	0.1	3	20
$3-5$	$0.2, 0.3, 0.4$	3	16
$6-9$	$0.5, 0.4, 0.6, 0.7$	4	16

.

Parameters for method SD2-LS1

- 10 frequencies with one recycle for a total 11 runs: κ = 0.09,0.1:0.1:0.5, 0.4 (recycled), 0.6, 0.7.
- Common to all runs: $N_{\text{IterMax}} = 60$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 1.0 \times 10^{-5} \ ; \ \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = 1.0 \times 10^{-3} \, .$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.09	3	24
2	0.1	3	20
$3, 4$	$0.2, 0.3$	3	18
5	0.4	4	16
$6-9$	$0.5, 0.4, 0.6, 0.7$	4	10

Remark 18. *These are parameters regarding Remark 13. To obtain scaled error position 0.08%, we can use two more frequencies: 0.8, 0.9, with corresponding parameters.*

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{Stag\ LS} = 1.0 \times 10^{-5} \ ; \ \epsilon_{Stag\ J} = 0.1 \ , \ \epsilon_{Stag\ Pos} 5.0 \times 10^{-4} \, .$$

Run	κ	***Order FSSL***	***Reference step length***
$10, 11$	$0.8, 0.9$	4	10

E.5 Twelve obstacles reconstruction: noisy data – IG1

Parameters for method SD1-LS3

- 8 frequencies with one recycle for a total 9 runs: κ = 0.09, 0.1 : 0.1 : 0.5, 0.4, 0.4 (recycled), 0.6, 0.7.
- Common to all runs: $N_{\text{IterMax}} = 70$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = \epsilon_{\text{Stag Pos}} = 1.0 \times 10^{-5} \quad , \quad \epsilon_{\text{Stag J}} = 0.1$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.09	3	1
2	0.1	3	24
$3, 4, 5$	$0.2, 0.3, 0.4$	3	18
$6, 7$	$0.5, 0.4$	4	16
$8, 9$	$0.6, 0.7$	4	10

Parameters for method SD2-LS3

- 8 frequencies with one recycle for a total 9 runs: $\kappa = 0.09$, $0.1 : 0.1 : 0.5$, 0.4, 0.4 (recycled), 0.6, 0.7.
- $N_{\text{IterMax}} = \begin{cases} 60 & \text{Run } 1-8 \\ 150 & \text{Run } 9 \end{cases}$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 1.0 \times 10^{-5} \quad , \quad \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 5.0 \times 10^{-4} & \text{Run 1, 2, 3} \\ 5.0 \times 10^{-5} & \text{Run 4–9} \end{cases}$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.09	3	24
2	0.1	3	20
3	0.2	3	18
$4,5$	$0.3, 0.4$	3	16
6	0.5	4	16
$7-9$	$0.4, 0.6, 0.7$	4	12

E.6 Twelve obstacles reconstruction: noisy data – IG2

Parameters for method SD1-LS3

- 8 frequencies with one recycle for a total 9 runs: $\kappa = 0.09$, $0.1 : 0.1 : 0.5$, 0.4, 0.4 (recycled), 0.6, 0.7.
- Common to all runs: $N_{\text{IterMax}} = 70$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 1.0 \times 10^{-5} \quad , \quad \epsilon_{\text{Stag Pos}} = 5.0 \times 10^{-5} \quad , \quad \epsilon_{\text{Stag J}} = 0.1$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.09	3	1
$2-5$	$0.1 : 0.4$	3	24
6	0.5	4	24
$7,8,9$	$0.4, 0.6, 0.7$	4	18

Parameters for method SD2-LS3

- 8 frequencies with one recycle for a total 9 runs: $\kappa = 0.09$, $0.1 : 0.1 : 0.5$, 0.4, 0.4 (recycled), 0.6, 0.7.
- Common to all runs: $N_{\text{IterMax}} = \begin{cases} 60 & \text{Run 1–8} \\ 150 & \text{Run 9} \end{cases}$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag LS}} = 1.0 \times 10^{-5} \quad , \quad \epsilon_{\text{Stag J}} = 0.1 \ , \ \epsilon_{\text{Stag Pos}} = \begin{cases} 5.0 \times 10^{-3} & \text{Run 1 - 5} \\ 5.0 \times 10^{-4} & \text{Run 6} \\ 5.0 \times 10^{-5} & \text{Run 7 - 9} \end{cases} .$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.09	3	24
2	0.1	3	20
3	0.2	3	18
$4, 5$	$0.3, 0.4$	3	16
6	0.5	4	16
$7-9$	$0.4, 0.6, 0.7$	4	10

E.7 Random twelve obstacles reconstruction: noise free

Parameters for method SD1-LS3

- $\mu = 1.0 \times 10^{-4}$, $\eta = 0.4$.
- Seven frequencies: $\kappa = 0.08$, $0.1 : 0.1 : 0.5$,
- Common to all runs: $N_{\text{IterMax}} = 150$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\nabla J} = \epsilon_{\text{Stag,LS}} = 1.0 \times 10^{-6} \;;\; \epsilon_{\text{Stag},J} = 0.1\,;$$

Run	κ	$\epsilon_{\text{Stag Pos}}$	**Reference step length**	
1	0.08	3	1.0×10^{-6}	30
2	0.09	3	1.0×10^{-6}	24
3	0.1	3	1.0×10^{-7}	20
4	0.2	3	1.0×10^{-7}	18
5	0.3	3	1.0×10^{-7}	18
6	0.4	3	1.0×10^{-7}	16
7	0.5	4	1.0×10^{-7}	16

E.8 Random twelve obstacles reconstruction: noisy data

Parameters for method SD1-LS3

- $\mu = 1.0 \times 10^{-4}$, $\eta = 0.4$.
- Ten frequencies: $\kappa = 0.08$, $0.1 : 0.1 : 0.8$,
- Common to all runs: $N_{\text{IterMax}} = 300$, $N_{\text{LS IterMax 1}} = 30$, $N_{\text{LS IterMax 2}} = 30$.
- Regarding the error tolerances and stagnation parameters,

$$\epsilon_J = \epsilon_{\text{Stag}} = 5.0 \times 10^{-2} \;;\; \epsilon_{\nabla\, J} = 5.0 \times 10^{-4} \;;\; \epsilon_{\text{Stag J}} = 0.1 \;;\; \epsilon_{\text{Stag Pos}} = 1.0 \times 10^{-8}\,.$$

Run	κ	**Order FSSL**	**Reference step length**
1	0.08	3	50
2	0.09	3	30
$3-6$	$0.1 : 0.1 : 0.4$	3	10
$7-10$	$0.5 : 0.1 : 0.8$	4	10

Contents

References

[1] H. Ammari, E. Iakovleva, and D. Lesselier, *A MUSIC algorithm for locating small inclusions buried in a half-space from the scattering amplitude at a fixed frequency*, Multiscale Modeling & Simulation, 3 (2005), pp. 597–628.

[2] H. Ammari and H. Kang, *Reconstruction of small inhomogeneities from boundary measurements*, Springer, 2004.

[3] X. Antoine, C. Geuzaine, and K. Ramdani, *Computational methods for multiple scattering at high frequency with applications to periodic structure calculations*, Wave Propagation in Periodic Media, Progress in Computational Physics, 1 (2010), pp. 73–107.

[4] G. Bao, K. Huang, P. Li, and H. Zhao, *A direct imaging method for inverse scattering using the generalized Foldy-Lax formulation*, Contemp. Math, 615 (2014), pp. 49–70.

[5] G. Bao, P. Li, J. Lin, and F. Triki, *Inverse scattering problems with multi-frequencies*, Inverse Problems, 31 (2015), p. 093001.

[6] H. Barucq, J. Chabassier, H. Pham, and S. Tordeux, *A study of the numerical robustness of single-layer method with Fourier basis for multiple obstacle scattering in homogeneous media*, Research Report RR-8988, Inria Bordeaux Sud-Ouest, Dec. 2016.

[7] H. Barucq, J. Chabassier, H. Pham, and S. Tordeux, *Numerical robustness of single-layer method with Fourier basis for multiple obstacle acoustic scattering in homogeneous media*, Wave Motion, (2017).

[8] C. Borges and L. Greengard, *Inverse obstacle scattering in two dimensions with multiple frequency data and multiple angles of incidence*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 280–298.

[9] O. M. Bucci, L. Crocco, T. Isernia, and V. Pascazio, *Inverse scattering problems with multifrequency data: reconstruction capabilities and solution strategies*, IEEE Transactions on Geoscience and Remote Sensing, 38 (2000), pp. 1749–1756.

[10] F. Cakoni and D. Colton, *Qualitative methods in inverse scattering theory: An introduction*, Springer Science & Business Media, 2005.

[11] D. P. Challa and M. Sini, *Inverse scattering by point-like scatterers in the Foldy regime*, Inverse Problems, 28 (2012), p. 125006.

[12] G. Chavent, *Nonlinear least squares for inverse problems: theoretical foundations and step-by-step guide for applications*, Springer Science & Business Media, 2010.

[13] D. Colton and R. Kress, *Inverse acoustic and electromagnetic scattering theory*, vol. 93, Springer Science & Business Media, 2012.

[14] D. Colton and B. Sleeman, *Uniqueness theorems for the inverse problem of acoustic scattering*, IMA journal of applied mathematics, 31 (1983), pp. 253–259.

[15] J. E. Dennis, Jr and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM review, 19 (1977), pp. 46–89.

[16] A. J. Devaney, E. A. Marengo, and F. K. Gruber, *Time-reversal-based imaging and inverse scattering of multiply scattering point targets*, The Journal of the Acoustical Society of America, 118 (2005), pp. 3129–3138.

[17] F. Faucher, *Contributions to seismic full waveform inversion for time harmonic wave equations: stability estimates, convergence analysis, numerical experiments involving large scale optimization algorithms*, PhD thesis, Université de Pau et Pays de l'Ardour., 2017.

[18] R. Fazli and M. Nakhkash, *An analytical approach to estimate the number of small scatterers in 2d inverse scattering problems*, Inverse Problems, 28 (2012), p. 075012.

[19] A. Fichtner, *Full seismic waveform modelling and inversion*, Springer Science & Business Media, 2011.

[20] K. Firouzi and B. T. Khuri-Yakub, *Localization of weak objects in reverberant fields using waveform inversion*, The Journal of the Acoustical Society of America, 142 (2017), pp. 1088–1097.

[21] A. Friedman and M. Vogelius, *Identification of small inhomogeneities of extreme conductivity by boundary measurements: a theorem on continuous dependence*, Archive for Rational Mechanics and Analysis, 105 (1989), pp. 299–326.

[22] F. Gesztesy and A. G. Ramm, *An inverse problem for point inhomogeneities*, Methods Func. Anal. Topology, 6 (2000), pp. 1–12.

[23] D. Gintides, *Local uniqueness for the inverse scattering problem in acoustics via the Faber–Krahn inequality*, Inverse Problems, 21 (2005), p. 1195.

[24] R. Griesmaier and C. Schmiedecke, *A multifrequency MUSIC algorithm for locating small inhomogeneities in inverse scattering*, Inverse Problems, 33 (2017), p. 035015.

[25] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pacific journal of Optimization, 2 (2006), pp. 35–58.

[26] F. Hettlich, *Fréchet derivatives in inverse obstacle scattering*, Inverse problems, 11 (1995), p. 371.

[27] ———, *Fréchet derivatives in inverse obstacle scattering*, Inverse Problems, 14 (1998), p. 209.

[28] K. Huang, P. Li, and H. Zhao, *An efficient algorithm for the generalized Foldy–Lax formulation*, Journal of Computational Physics, 234 (2013), pp. 376–398.

[29] V. Isakov, *Inverse problems for partial differential equations*, vol. 127, Springer, 2006.

[30] M. Kern, *Numerical Methods for Inverse Problems*, John Wiley & Sons, 2016.

[31] R. Kleinman and P. den Berg, *Two-dimensional location and shape reconstruction*, Radio Science, 29 (1994), pp. 1157–1169.

[32] A. E. Kolesov, M. V. Klibanov, L. H. Nguyen, D.-L. Nguyen, and N. T. Thanh, *Single measurement experimental data for an inverse medium problem inverted by a multi-frequency globally convergent numerical method*, Applied Numerical Mathematics, (2017).

[33] R. Kress, *Newton's method for inverse obstacle scattering meets the method of least squares*, Inverse Problems, 19 (2003), p. S91.

[34] ——, *Uniqueness and numerical methods in inverse obstacle scattering*, in Journal of Physics: Conference Series, vol. 73, IOP Publishing, 2007.

[35] R. Kress and K.-M. Lee, *A second degree newton method for an inverse obstacle scattering problem*, Journal of Computational Physics, 230 (2011), pp. 7661–7669.

[36] R. Kress and W. Rundell, *A quasi-newton method in inverse obstacle scattering*, Inverse Problems, 10 (1994), p. 1145.

[37] ——, *Inverse obstacle scattering using reduced data*, SIAM Journal on Applied Mathematics, 59 (1998), pp. 442–454.

[38] D. R. Luke and R. Potthast, *The no response test—a sampling method for inverse scattering problems*, SIAM Journal on Applied Mathematics, 63 (2003), pp. 1292–1312.

[39] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Transactions on Mathematical Software (TOMS), 20 (1994), pp. 286–307.

[40] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[41] R. Potthast, *Fréchet differentiability of the solution to the acoustic neumann scattering problem with respect to the domain*, Journal of Inverse and Ill-Posed Problems, 4 (1996), pp. 67–84.

[42] R. G. Pratt, *Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model*, Geophysics, 64 (1999), pp. 888–901.

[43] E. Sincich and M. Sini, *Local stability for soft obstacles by a single measurement*, Inverse Problems and Imaging, 2 (2008), pp. 301–315.

[44] M. Sini and N. T. Thành, *Regularized recursive newton-type methods for inverse scattering problems using multifrequency measurements*, ESAIM: Mathematical Modelling and Numerical Analysis, 49 (2015), pp. 459–480.

[45] P. Van Den Berg, R. Kleinman, et al., *Gradient methods in inverse acoustic and electromagnetic scattering*, IMA Volumes in Mathematics and its Applications, 92 (1997), pp. 173–194.

[46] J. Virieux and S. Operto, *An overview of full waveform inversion in exploration geophysics*, Geophysics, 74 (2009), pp. WCC1–WCC26.

[47] F. Yaman, V. G. Yakhno, and R. Potthast, *A survey on inverse problems for applied sciences*, Mathematical problems in engineering, 2013 (2013).

[48] X. Zhang, S. L. Broschat, and P. J. Flynn, *A numerical study of conjugate gradient directions for an ultrasound inverse problem*, Journal of Computational Acoustics, 12 (2004), pp. 587–604.

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399