Waveform inversion via reduced order modeling

Liliana Borcea*, Josselin Garnier†, Alexander V. Mamonov‡ and Jörn Zimmerling*

ABSTRACT

We introduce a novel approach to waveform inversion, based on a data driven reduced order model (ROM) of the wave operator. The presentation is for the acoustic wave equation, but the approach can be extended to elastic or electromagnetic waves. The data are time resolved measurements of the pressure wave at the sensors in an active array, which probe the unknown medium with pulses and measure the generated waves. The ROM depends nonlinearly on the data but it can be constructed from them using numerical linear algebra methods. We show that the ROM can be used for the inverse problem of velocity estimation. While the full-waveform inversion approach of nonlinear least-squares data fitting is challenging without low frequency information, due to multiple minima of the objective function, the minimization of the ROM misfit function has a better behavior, even for a poor initial guess. In fact, the ROM misfit function is demonstrably a convex function for low-dimensional parametrizations of the unknown velocity. We give the construction of the ROM, introduce the inversion approach based on the ROM misfit and assess its performance with numerical simulations.

INTRODUCTION

We study the inverse problem of velocity estimation from reflection data gathered by an array of m sensors. The proposed methodology applies to any linear wave equation, for scalar (sound) or vectorial (electromagnetic or elastic) waves, but for simplicity we work with the acoustic wave equation in a medium with constant density and unknown wave speed, aka velocity $c(\boldsymbol{x})$.

Let $p^{(s)}(t,\boldsymbol{x})$ model the pressure wave generated by the s^{th} sensor, for $s = 1,\dots,m$. It satisfies the initial value problem

$$\left[\partial_t^2 - c^2(\boldsymbol{x})\Delta\right]p^{(s)}(t,\boldsymbol{x}) = f'(t)\theta(\boldsymbol{x}-\boldsymbol{x}_s), \quad t \in \mathbb{R}, \quad (1)$$

$$p^{(s)}(t,\boldsymbol{x}) = 0, \quad t < -t_f, \quad (2)$$

for $\boldsymbol{x} \in \Omega$, a simply connected domain in dimension $d = 2$ or 3, with boundary $\partial\Omega$. This domain arises from the mathematical truncation of $\mathbb{R}^d$, since over the finite duration T of the measurements, the waves are not affected by the medium at distances exceeding $T \max_{\boldsymbol{x}} c(\boldsymbol{x})$. Thus, we can set any homogeneous boundary conditions: for example Dirichlet or a combination of Dirichlet and Neumann on disjoint parts of $\partial\Omega$. We refer to these henceforth as "the homogeneous boundary conditions".

The sensors in the array are assumed identical, and are modeled by a function $\theta(\boldsymbol{x})$, with small support around the origin $\mathbf{0}$. For example, $\theta(\boldsymbol{x})$ may be equal to one in a small ball centered at $\mathbf{0}$ and equal to zero everywhere else. The right-hand side in (1) models the excitation from the s^{th} sensor, which emits the probing pulse $f(t)$, supported in the time interval $(-t_f, t_f)$. Prior to the excitation the medium is quiescent, as stated in the initial condition (2).

The inverse problem is: Find the velocity $c(\boldsymbol{x})$ from the measurements

$$\mathcal{M}^{(r,s)}(t) = \int_\Omega d\boldsymbol{x}\, \theta(\boldsymbol{x}-\boldsymbol{x}_r)p^{(s)}(t,\boldsymbol{x}), \quad (3)$$

for $s, r = 1,\dots,m$ and $t \in (0,T)$. Here we assume that each sensor can both emit and record.

Common velocity estimation approaches are: Travel time tomography (Dines and Lytle (1979)) and its more general version studied in the mathematics community (Stefanov et al. (2019)); Linearized, aka Born inversion (Clayton and Stolt (1981)); Migration velocity analysis (Symes and Carazzone (1991); Sava and Biondi (2004)) and Full-waveform inversion (Tarantola (1984); Virieux and Operto (2009)). The first three are based on assumptions like: the velocity changes slowly on the scale of the

*University of Michigan, Department of Mathematics, Ann Arbor, MI 48109-1043. †Centre de Mathématiques Appliquées, Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau Cedex, France. ‡University of Houston, Department of Mathematics, Houston, TX 77204-3008.

wavelength (for travel time tomography), or the velocity variations are small (for Born inversion) or there is separation of scales between the smooth components of the velocity and the rough part that gives the reflectivity of the medium (for migration). Full-waveform inversion (FWI) circumvents such assumptions. It is a PDE constrained optimization that fits the data with its model prediction in the L^2 (least-squares) sense. The increase in computing power has lead to growing interest in FWI, but there is a fundamental impediment, which manifests especially for high-frequency data: The objective function is nonconvex in the smooth component of $c(\boldsymbol{x})$ even in the absence of noise (Gauthier et al. (1986); Santosa and Symes (1989)) and displays numerous local minima. This so-called "cycle skipping" issue makes any gradient based, i.e., local optimization algorithm unlikely to succeed, in the absence of an accurate starting guess (Virieux and Operto (2009)).

There are several approaches to mitigate cycle skipping: Multiscale methods pursue a good starting guess by inverting first very low frequency data (Bunks et al. (1995)). However, such data may not be available and there is no guarantee that what seems a reasonable starting guess will not create cycle skipping issues for high-frequency data. Extended modeling approaches (Symes (2008)) like the differential semblance method (Symes and Carazzone (1991); Symes and Kern (1994)) and the source-receiver extension method (Huang et al. (2017)), introduce in a systematic way additional degrees of freedom in the optimization and then use some objective function to drive the extended model toward a velocity estimate. There are also approaches that use a better alternative than the L^2 norm for measuring the data misfit (Brossier et al. (2010); Bozdağ et al. (2011)). A prominent alternative is the optimal transport (Wasserstein) metric proposed and analyzed for seismic inversion in (Engquist and Froese (2014); Yang et al. (2018)).

We introduce a different approach to velocity estimation, based on a data driven reduced order model (ROM) of the wave operator. The mapping between the measurements (3) and the ROM is nonlinear and yet, it can be calculated efficiently with methods from numerical linear algebra. The main point of the paper is that the objective function given by the ROM misfit has better behavior than the FWI objective function, so optimization methods can converge even for a poor initial guess.

There is an evergrowing list of data driven ROM approaches to operator inference and dynamical system identification (Brunton et al. (2016); Peherstorfer and Willcox (2016)). However, they require data that are not available in our inverse problem, meaning that they assume knowledge of the state of the system, the wave $p^{(s)}(t,\boldsymbol{x})$ in our case, at a finite set of time instants and for all $\boldsymbol{x} \in \Omega$. We only have the measurements (3) of the wave.

The first sensor array data driven ROM for wave propagation was introduced and used in (Druskin et al. (2016)) in one dimension and in (Borcea et al. (2018, 2019, 2020)) in higher dimensions. The ROM in these studies is not for the wave operator, but for the "propagator" operator which maps the wave field from one instant to the next one, on a uniform time grid. The ROM propagator has proved useful for imaging the reflectivity of a medium (Druskin et al. (2018); Borcea et al. (2020)). In this paper we introduce another ROM, for the wave operator, which is better suited for velocity estimation. In fact, we demonstrate with explicit computations, carried out for a low-dimensional velocity model, that the wave operator ROM misfit objective function is convex. This is not the case for the FWI and ROM propagator misfit objective functions, computed for the same velocity model. For high-dimensional models, where it is not possible to display the objective function, we show via numerical simulations that the wave operator ROM-based inversion converges to a good estimate of $c(\boldsymbol{x})$, even for a poor initial guess, whereas FWI does not.

THEORY

We begin with the definition and construction of the two data driven ROMs: for the wave propagator operator and the wave operator. These two ROMs are related and they capture the wave propagation in the unknown medium in complementary ways. Then, we give the ROM-based inversion algorithm and describe its implementation. The discussion in this section is for noiseless data. We consider later the regularization of the data driven ROM construction, where the two ROMs are used in conjunction to deal with noisy data.

Setup for the ROM construction

We assume throughout the paper that the pulse $f(t)$ is given by

$$f(t) = \varphi(t) \star_t \varphi(-t), \tag{4}$$

in terms of a wavelet $\varphi(t)$ with compact support contained in $(-t_f/2, t_f/2)$, where $\star_t$ denotes convolution in time. This can be achieved in practice if the probing pulse emitted by the sensors is actually $\varphi(t)$, and if the form of φ is known or can be estimated (Pratt (1999)). Then, the measured wave convolved with $\varphi(-t)$ is the same as the solution of equation (1) evaluated at the sensors, with $f(t)$ given in (4).

Assumption (4) is a technical requirement used in the ROM construction. In particular, we use that $f(t)$ is an even function, with non-negative Fourier transform

$$\widehat{f}(\omega) = |\widehat{\varphi}(\omega)|^2 = \int_{\mathbb{R}} dt\, f(t) e^{i\omega t} \geq 0, \tag{5}$$

that is analytic by the Paley-Wiener-Schwartz theorem.

It is convenient for the theory to have a self-adjoint wave operator. Thus, we write the wave equation (1) in terms of $p^{(s)}(t,\boldsymbol{x})/c(\boldsymbol{x})$, so that the operator $-c^2(\boldsymbol{x})\Delta$ becomes

$$\mathcal{A} = -c(\boldsymbol{x})\Delta\big[c(\boldsymbol{x})\cdot\big]. \tag{6}$$

This is just a mathematical similarity transformation and assuming that the velocity is known and constant in the support of each $\theta(\boldsymbol{x}-\boldsymbol{x}_r)$, for $r=1,\ldots,m$, we can easily transform the measurements (3) of the pressure to those of $p^{(s)}(t,\boldsymbol{x})/c(\boldsymbol{x})$.

We work with the even in time wave

$$\mathrm{w}^{(s)}(t,\boldsymbol{x}) = \frac{\left[p^{(s)}(t,\boldsymbol{x})+p^{(s)}(-t,\boldsymbol{x})\right]}{c(\boldsymbol{x})}, \tag{7}$$

where we note that $p^{(s)}(-t,\boldsymbol{x})=0$ for $t>t_f$, due to the initial condition (2). We call henceforth "the data" the $m\times m$ matrix $\boldsymbol{D}(t)$, with entries

$$\begin{aligned} D^{(r,s)}(t) &= \int_\Omega d\boldsymbol{x}\,\frac{\theta(\boldsymbol{x}-\boldsymbol{x}_r)}{c(\boldsymbol{x}_r)}\mathrm{w}^{(s)}(t,\boldsymbol{x}) \\ &= \frac{\mathcal{M}^{(r,s)}(t)+\mathcal{M}^{(r,s)}(-t)}{c(\boldsymbol{x}_r)c(\boldsymbol{x}_s)}, \end{aligned} \tag{8}$$

for $s,r=1,\ldots,m$. These can be obtained from the measurements (3), because the velocity is known at the sensors and $\mathcal{M}^{(r,s)}(-t)=0$ for $t>t_f$, by equation (2). Moreover, at $t\in(-t_f,t_f)$ the wave $p^{(s)}(t,\boldsymbol{x})$ is affected only by the medium within an order $c(\boldsymbol{x}_s)t_f$ distance from the s^{th} sensor. Thus, we can compute it and therefore $\mathcal{M}^{(r,s)}(-t)$ by solving the wave equation in the vicinity of $\boldsymbol{x}_s$, where we know the medium.

The ROM propagator is computed from $2n$ equidistant time samples of the matrices (8), denoted henceforth by

$$\boldsymbol{D}_j = \boldsymbol{D}(j\tau), \quad j=0,\ldots,2n-1. \tag{9}$$

The ROM of the wave operator is computed from

$$\boldsymbol{D}_j \text{ and } \ddot{\boldsymbol{D}}_j = \partial_t^2\boldsymbol{D}(j\tau), \quad j=0,\ldots,2n-2. \tag{10}$$

The data sampling interval τ should be chosen according to the Nyquist sampling rate for the essential* frequency content of $f(t)$ and the derivatives in (10) can be obtained from the measurements using the Fourier transform (see Appendix C).

The mathematical transformations above allow us to write the wave (7) in the following form derived in (Borcea et al., 2020, Appendix A)

$$\mathrm{w}^{(s)}(t,\boldsymbol{x}) = \widehat{f}^{\frac12}\big(\sqrt{\mathcal{A}}\big)u^{(s)}(t,\boldsymbol{x}), \tag{11}$$

where $\widehat{f}^{\frac12}(\omega)=|\widehat{\varphi}(\omega)|$ by equation (5) and

$$u^{(s)}(t,\boldsymbol{x}) = \cos\big(t\sqrt{\mathcal{A}}\big)u_0^{(s)}(\boldsymbol{x}), \tag{12}$$

solves the initial value problem

$$\big[\partial_t^2+\mathcal{A}\big]u^{(s)}(t,\boldsymbol{x})=0, \quad t>0,\ \boldsymbol{x}\in\Omega, \tag{13}$$

$$u^{(s)}(0,\boldsymbol{x})=u_0^{(s)}(\boldsymbol{x}), \ \ \partial_t u^{(s)}(0,\boldsymbol{x})=0, \quad \boldsymbol{x}\in\Omega, \tag{14}$$

with the homogeneous boundary conditions at $\partial\Omega$. The initial state of this wave is given by

$$u_0^{(s)}(\boldsymbol{x}) = \widehat{f}^{\frac12}\big(\sqrt{\mathcal{A}}\big)\frac{\theta(\boldsymbol{x}-\boldsymbol{x}_s)}{c(\boldsymbol{x}_s)}, \tag{15}$$

and it is shown in (Borcea et al., 2020, Appendix A) that it is supported in a ball centered at $\boldsymbol{x}_s$, with radius of order $c(\boldsymbol{x}_s)t_f$. Moreover, it can be computed by solving the wave equation near $\boldsymbol{x}_s$, where the medium is known.

In equations (11)-(12) and (15) we use the standard definition of a function F of the self-adjoint operator $\mathcal{A}$ (with $F:\mathbb{R}\to\mathbb{R}$), based on its spectral decomposition (McLean, 2000, Theorem 4.12). Specifically, if we denote by $\{\lambda_l,\ l\geq 1\}$ the eigenvalues of $\mathcal{A}$, which are positive, listed in increasing order and satisfying $\lambda_l \overset{l\to\infty}{\longrightarrow} \infty$, and by $\{y_l(\boldsymbol{x}),\ l\geq 1\}$ the corresponding eigenfunctions, which form an orthonormal basis of $L^2(\Omega)$ with the homogeneous boundary conditions, then

$$F\big(\mathcal{A}\big)\phi(\boldsymbol{x}) = \sum_{l=1}^\infty F(\lambda_l)y_l(\boldsymbol{x})\int_\Omega d\boldsymbol{x}' y_l(\boldsymbol{x}')\phi(\boldsymbol{x}'),$$

for any function $\phi(\boldsymbol{x})$ in the domain of $\mathcal{A}$. Since the eigenvalues of $\mathcal{A}$ are strictly positive, we work with two analytic functions of $\mathcal{A}$, $F(\mathcal{A})=\widehat{f}^{\frac12}\big(\sqrt{\mathcal{A}}\big)$ and $\cos\big(t\sqrt{\mathcal{A}}\big)$, which commute. This is why we could factor out the wave as in equation (11).

An important observation used in the ROM construction is that the data matrices (9)-(10) can be expressed in a symmetric inner product form. Indeed, substituting (11) and (12) into (8) and then using definition (15) we get

$$\begin{aligned} D_j^{(r,s)} &= \int_\Omega d\boldsymbol{x}\,\frac{\theta(\boldsymbol{x}-\boldsymbol{x}_r)}{c(\boldsymbol{x}_r)}\widehat{f}^{\frac12}\big(\sqrt{\mathcal{A}}\big)u^{(s)}(j\tau,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\,u_0^{(r)}(\boldsymbol{x})u^{(s)}(j\tau,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\,u_0^{(r)}(\boldsymbol{x})\cos\big(j\tau\sqrt{\mathcal{A}}\big)u_0^{(s)}(\boldsymbol{x}), \end{aligned} \tag{16}$$

and

$$\begin{aligned} \ddot{D}_j^{(r,s)} &= \int_\Omega d\boldsymbol{x}\,\frac{\theta(\boldsymbol{x}-\boldsymbol{x}_r)}{c(\boldsymbol{x}_r)}\widehat{f}^{\frac12}\big(\sqrt{\mathcal{A}}\big)\partial_t^2 u^{(s)}(j\tau,\boldsymbol{x}) \\ &= -\int_\Omega d\boldsymbol{x}\,u_0^{(r)}(\boldsymbol{x})\mathcal{A}u^{(s)}(j\tau,\boldsymbol{x}) \\ &= -\int_\Omega d\boldsymbol{x}\,u_0^{(r)}(\boldsymbol{x})\mathcal{A}\cos\big(j\tau\sqrt{\mathcal{A}}\big)u_0^{(s)}(\boldsymbol{x}), \end{aligned} \tag{17}$$

for $r,s=1,\ldots,m$ and $j=0,\ldots,2n-2$. Here we used that $\widehat{f}^{\frac12}\big(\sqrt{\mathcal{A}}\big)$ is self-adjoint and we note that $\mathcal{A}$ and $\cos\big(j\tau\sqrt{\mathcal{A}}\big)$ commute.

To avoid cumbersome notation with multiple indexes, we use henceforth block algebra. Thus, we gather all the waves (12) evaluated at the time instant $j\tau$ into the m-dimensional row vector field called a snapshot

$$\boldsymbol{u}_j(\boldsymbol{x}) = \big(u^{(1)}(j\tau,\boldsymbol{x}),\ldots,u^{(m)}(j\tau,\boldsymbol{x})\big), \tag{18}$$

and obtain from (12) that

$$\boldsymbol{u}_j(\boldsymbol{x}) = \cos\big(j\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0(\boldsymbol{x}), \quad j\geq 0. \tag{19}$$

By (16-17) the data matrices can be written as

$$\boldsymbol{D}_j = \langle\boldsymbol{u}_0,\boldsymbol{u}_j\rangle = \langle\boldsymbol{u}_0,\cos\big(j\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0\rangle, \tag{20}$$

$$\ddot{\boldsymbol{D}}_j = -\langle\boldsymbol{u}_0,\mathcal{A}\boldsymbol{u}_j\rangle, \tag{21}$$

*The Fourier transform $\widehat{f}(\omega)$ of the pulse is small outside a frequency interval called the "essential frequency content" of $f(t)$. The largest frequency in this interval is referred to as the "essential Nyquist frequency".

where $\langle \boldsymbol{\phi}, \boldsymbol{\psi} \rangle = \int_\Omega d\boldsymbol{x}\, \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x})$ denotes the integral of the outer product of any functions $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\psi}(\boldsymbol{x})$ with values in $\mathbb{R}^{1\times m}$ and superscript T stands for the transpose.

The ROM of the propagator operator

The expression (19) of the snapshots of the wave is fundamental to the construction of the ROMs for at least two reasons: First, we use it in conjunction with the trigonometric identity of the cosine

$$\cos(\alpha)\cos(\beta) = \frac{1}{2}\left[\cos(\alpha+\beta) + \cos(\alpha-\beta)\right], \tag{22}$$

to compute the ROMs from the data. Second, it allows writing the wave propagation as an exact, discrete in time, evolution equation driven by the propagator operator

$$\mathcal{P} = \cos\left(\tau\sqrt{\mathcal{A}}\right). \tag{23}$$

Indeed, the snapshots evolve from the known $\boldsymbol{u}_0(\boldsymbol{x})$ according to

$$\begin{aligned}
\boldsymbol{u}_{j+1}(\boldsymbol{x}) &= \cos\left((j+1)\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0(\boldsymbol{x}) \\
&= \Big[2\cos\left(\tau\sqrt{\mathcal{A}}\right)\cos\left(j\tau\sqrt{\mathcal{A}}\right) - \cos\left((j-1)\tau\sqrt{\mathcal{A}}\right)\Big]\boldsymbol{u}_0(\boldsymbol{x}) \\
&= 2\mathcal{P}\boldsymbol{u}_j(\boldsymbol{x}) - \boldsymbol{u}_{j-1}(\boldsymbol{x}), \quad j \geq 0,
\end{aligned} \tag{24}$$

where we let $\boldsymbol{u}_{-1}(\boldsymbol{x}) = \boldsymbol{u}_1(\boldsymbol{x})$ and used identity (22).

We now review briefly from (Borcea et al. (2020)) the construction of the ROM propagator and its properties relevant to this paper. The ROM is obtained from the Galerkin projection of (24) onto the nm-dimensional space[†]

$$\mathbb{S} = \text{span}\{\boldsymbol{u}_0(\boldsymbol{x}), \ldots, \boldsymbol{u}_{n-1}(\boldsymbol{x})\}, \tag{25}$$

spanned by the first n snapshots. If we gather these snapshots into the nm-dimensional row vector field

$$\boldsymbol{U}(\boldsymbol{x}) = \big(\boldsymbol{u}_0(\boldsymbol{x}), \ldots, \boldsymbol{u}_{n-1}(\boldsymbol{x})\big) \in \mathbb{R}^{1\times nm}, \tag{26}$$

then the Galerkin approximation of the wave snapshots is

$$\boldsymbol{u}_j^{\text{GAL}}(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{g}_j, \quad j \geq 0, \tag{27}$$

where $\boldsymbol{g}_j \in \mathbb{R}^{nm\times m}$ are matrices of Galerkin coefficients, calculated so that when substituting (27) into (24), the residual term is orthogonal to $\mathbb{S}$. This means explicitly

$$\boldsymbol{M}(\boldsymbol{g}_{j+1} + \boldsymbol{g}_{j-1}) = 2\boldsymbol{S}\boldsymbol{g}_j, \quad j \geq 0, \tag{28}$$

where we introduced the Galerkin mass matrix

$$\boldsymbol{M} = \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x}) \in \mathbb{R}^{nm\times nm} \tag{29}$$

and the stiffness matrix

$$\boldsymbol{S} = \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\mathcal{P}\boldsymbol{U}(x) \in \mathbb{R}^{nm\times nm}. \tag{30}$$

[†]The space $\mathbb{S}$ is nm-dimensional because the snapshots $\boldsymbol{u}_j(\boldsymbol{x})$, for $j = 0, \ldots, n-1$, are assumed linearly independent.

Galerkin projections on the space spanned by snapshots are routinely used in ROM constructions (Brunton and Kutz (2019); Hesthaven et al. (2016)). There are at least three main differences in our construction: First, our goal is not to compress the information contained in a large approximation space, defined by possibly redundant snapshots, via some method like the proper orthogonal decomposition (Kunisch and Volkwein (2010)), as is common in model order reduction. Our snapshots must be linearly independent, so the ROM propagator is an $nm \times nm$ matrix given by the projection of the propagator operator $\mathcal{P}$ on the nm-dimensional space $\mathbb{S}$ defined in (25). Second, we compute the ROM solely from the data matrices (9), without knowing the space $\mathbb{S}$. Third, the discrete evolution equation (24) is exact, i.e., it does not use some finite difference approximation of the operator ∂_t^2, as is typical in model order reduction (Herkt et al. (2013)). Consequently, our ROM has good approximation properties.

The data driven ROM construction begins with the observation that the $m \times m$ blocks of the mass and stiffness matrices $\boldsymbol{M}$ and $\boldsymbol{S}$ can be computed as follows

$$\begin{aligned}
\boldsymbol{M}_{i,j} &= \langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = \langle \cos\left(i\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0, \cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0 \rangle \\
&= \langle \boldsymbol{u}_0, \cos\left(i\tau\sqrt{\mathcal{A}}\right)\cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0 \rangle \\
&= \frac{1}{2}\langle \boldsymbol{u}_0, \big[\cos\left((i+j)\tau\sqrt{\mathcal{A}}\right) + \cos\left(|i-j|\tau\sqrt{\mathcal{A}}\right)\big]\boldsymbol{u}_0 \rangle \\
&= \frac{1}{2}\left(\boldsymbol{D}_{i+j} + \boldsymbol{D}_{|i-j|}\right),
\end{aligned} \tag{31}$$

and

$$\begin{aligned}
\boldsymbol{S}_{i,j} &= \langle \boldsymbol{u}_i, \mathcal{P}\boldsymbol{u}_j \rangle = \frac{1}{2}\langle \boldsymbol{u}_i, \boldsymbol{u}_{j+1} + \boldsymbol{u}_{j-1} \rangle \\
&= \frac{1}{4}\left(\boldsymbol{D}_{i+j+1} + \boldsymbol{D}_{|i-j-1|} + \boldsymbol{D}_{|i+j-1|} + \boldsymbol{D}_{|i-j+1|}\right),
\end{aligned} \tag{32}$$

for $i, j = 0, \ldots, n-1$. Here we used the identity (22) and the self-adjointness of $\mathcal{A}$.

By construction, the approximation (27) is exact at the first n time instants, meaning that the first n Galerkin coefficients matrices are simply the $nm \times m$ column blocks $\{\boldsymbol{e}_j\}_{j=0}^{n-1}$ of the $nm \times nm$ identity matrix $\boldsymbol{I}_{nm}$, i.e.,

$$\boldsymbol{g}_j = \boldsymbol{e}_j, \quad j = 0, \ldots, n-1. \tag{33}$$

Therefore, we can compute everything in the Galerkin equation (28) just from the data matrices $\{\boldsymbol{D}_j\}_{j=0}^{2n-1}$.

To get the ROM, we transform equation (28) to a form that captures the causal progression of the wavefront away from the sensors. This transformation is obtained using the square root of the mass matrix $\boldsymbol{M}$, which is symmetric and positive definite by definition (29). We take the square root using the block Cholesky factorization (Golub and Van Loan (2013)):

$$\boldsymbol{M} = \boldsymbol{R}^T\boldsymbol{R}, \tag{34}$$

where $\boldsymbol{R}$ is an $nm \times nm$ block upper triangular matrix, with $m \times m$ sized blocks. The upper triangular structure is key here, as we can relate the indices j of the $nm \times m$

column blocks of $\boldsymbol{R}$ to the time instants of the snapshots contained in $\boldsymbol{U}(\boldsymbol{x})$ and the indices i of the $m \times nm$ row blocks of $\boldsymbol{R}$ to the depth reached by the wave front. Indeed, defining the ROM snapshots by

$$\boldsymbol{u}_j^{\text{ROM}} = \boldsymbol{R}\boldsymbol{g}_j, \tag{35}$$

we get

$$\boldsymbol{u}_j^{\text{ROM}} = \boldsymbol{R}\boldsymbol{g}_j = \boldsymbol{R}\boldsymbol{e}_j, \quad j = 0, \dots, n-1, \tag{36}$$

which is an $nm \times m$ block column matrix, with nonzero entries contained in the row blocks indexed by $i = 0, \dots, j$. Thus, the j^{th} row block index is the algebraic analogue of the depth of the wavefront at time instant $j\tau$.

The evolution of the ROM snapshots is governed by the following equation, obtained by multiplying (28) on the left by $\boldsymbol{R}^{-T} = (\boldsymbol{R}^T)^{-1}$,

$$\boldsymbol{u}_{j+1}^{\text{ROM}} = 2\boldsymbol{\mathcal{P}}^{\text{ROM}}\boldsymbol{u}_j^{\text{ROM}} - \boldsymbol{u}_{j-1}^{\text{ROM}}, \quad j \geq 0. \tag{37}$$

This equation looks like (24). The initial ROM snapshot $\boldsymbol{u}_0^{\text{ROM}}$ has nonzero entries only in the first row block, thus capturing the spatial support of the true initial snapshot $\boldsymbol{u}_0(\boldsymbol{x})$ near the sensor. We also have $\boldsymbol{u}_{-1}^{\text{ROM}} = \boldsymbol{u}_1^{\text{ROM}}$. The ROM propagator is defined by

$$\boldsymbol{\mathcal{P}}^{\text{ROM}} = \boldsymbol{R}^{-T}\boldsymbol{S}\boldsymbol{R}^{-1}. \tag{38}$$

It is an $nm \times nm$ symmetric matrix with block-tridiagonal structure (Borcea et al., 2020, Appendix C).

To see that $\boldsymbol{\mathcal{P}}^{\text{ROM}}$ is in fact the projection of the operator $\mathcal{P}$, let us introduce the orthonormal basis of the approximation space $\mathbb{S}$, obtained via the block Gram-Schmidt orthogonalization of the components of $\boldsymbol{U}(\boldsymbol{x})$

$$\boldsymbol{U}(\boldsymbol{x}) = \boldsymbol{V}(\boldsymbol{x})\boldsymbol{R}. \tag{39}$$

The basis functions are in the nm-dimensional row vector field $\boldsymbol{V}(\boldsymbol{x})$, satisfying

$$\boldsymbol{V}(\boldsymbol{x})\boldsymbol{e}_j \in \text{span}\{\boldsymbol{u}_0(\boldsymbol{x}), \dots, \boldsymbol{u}_j(\boldsymbol{x})\}, \tag{40}$$

for $j = 0, \dots, n-1$ and

$$\int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{I}_{nm}. \tag{41}$$

In equation (39) we can take the same block upper triangular matrix $\boldsymbol{R}$ as in (34), because

$$\begin{aligned}\boldsymbol{M} &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x}) \\ &= \boldsymbol{R}^T \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x})\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{R},\end{aligned}$$

and the block Cholesky factorization obtained with (Druskin et al., 2018, Algorithm 5.2) is unique.

Substituting (39) into definition (30) of $\boldsymbol{S}$, we get the projection result

$$\begin{aligned}\boldsymbol{\mathcal{P}}^{\text{ROM}} &= \boldsymbol{R}^{-T} \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\mathcal{P}\boldsymbol{U}(\boldsymbol{x})\boldsymbol{R}^{-1} \\ &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\mathcal{P}\boldsymbol{V}(\boldsymbol{x}).\end{aligned} \tag{42}$$

We also note that the ROM snapshots are, from (35),

$$\begin{aligned}\boldsymbol{u}_j^{\text{ROM}} &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x})\boldsymbol{g}_j \\ &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{u}_j^{\text{GAL}}(\boldsymbol{x}),\end{aligned} \tag{43}$$

for $j = 0, \dots, n-1$.

Finally, we recall from (Borcea et al., 2020, Appendix B) that the ROM interpolates the data matrices used to construct it:

$$\boldsymbol{D}_j = \langle \boldsymbol{u}_0, \boldsymbol{u}_j \rangle = (\boldsymbol{u}_0^{\text{ROM}})^T\boldsymbol{u}_j^{\text{ROM}}, \quad j = 0, \dots, 2n-1. \tag{44}$$

The ROM of the wave operator

If we had an approximation of the operator $\mathcal{A}$, given by its projection on some known space, we would get a matrix with a simple (quadratic) dependence on $c(\boldsymbol{x})$, by definition (6). The ROM of the wave operator described in this section, and used later for velocity estimation, is the approximation of $\mathcal{A}$ on $\mathbb{S}$, an orthogonal projection

$$\begin{aligned}\boldsymbol{\mathcal{A}}^{\text{ROM}} &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\mathcal{A}\boldsymbol{V}(\boldsymbol{x}) \\ &= -\int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})c(\boldsymbol{x})\Delta\big[c(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x})\big].\end{aligned} \tag{45}$$

Note that $\mathbb{S}$ depends on $c(\boldsymbol{x})$ and it is difficult to quantify the mapping $c(\boldsymbol{x}) \mapsto \boldsymbol{V}(\boldsymbol{x})$ except for special cases discussed in (Borcea et al., 2021, Appendix A). Thus, we cannot compute analytically the Fréchet derivatives of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ with respect to $c(\boldsymbol{x})$ to study the convexity of the objective function. Nevertheless, we can expect that for a rich enough space $\mathbb{S}$, the approximation $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ will inherit the simple velocity dependence of $\mathcal{A}$. This is why we use it for velocity estimation.

Let us explain how to compute $\boldsymbol{\mathcal{A}}^{\text{ROM}}$. It is difficult to obtain it from $\boldsymbol{\mathcal{P}}^{\text{ROM}}$, which is according to (42) the projection of the operator $\cos\big(\tau\sqrt{\mathcal{A}}\big)$ on $\mathbb{S}$. In particular, we cannot approximate $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ by the matrix $\tau^{-2}\big[\arccos(\boldsymbol{\mathcal{P}}^{\text{ROM}})\big]^2$, calculated using the eigenvalues and eigenvectors of $\boldsymbol{\mathcal{P}}^{\text{ROM}}$, because these are only the Ritz approximation of the spectrum of $\mathcal{P}$ (Golub and Van Loan, 2013, Chapter 10). We compute instead $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ directly from the data matrices (10), as we now show.

The Galerkin approximation of the $m-$ dimensional row wave field $\boldsymbol{u}(t,\boldsymbol{x}) = \big(u^{(1)}(t,\boldsymbol{x}), \dots, u^{(m)}(t,\boldsymbol{x})\big)$ in the approximation space (25) has the form

$$\widetilde{\boldsymbol{u}}^{\text{GAL}}(t,\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})\widetilde{\boldsymbol{g}}(t), \tag{46}$$

where $\widetilde{\boldsymbol{g}}(t) \in \mathbb{R}^{nm\times m}$ is the time-dependent matrix of Galerkin coefficients calculated so that $\big(\partial_t^2 + \mathcal{A}\big)\widetilde{\boldsymbol{u}}^{\text{GAL}}(t,\boldsymbol{x})$ is orthogonal to $\mathbb{S}$. This gives, explicitly

$$\boldsymbol{M}\partial_t^2\widetilde{\boldsymbol{g}}(t) + \widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{g}}(t) = 0, \quad t > 0, \tag{47}$$

with $\widetilde{\boldsymbol{g}}(0) = \boldsymbol{e}_0$, $\partial_t\widetilde{\boldsymbol{g}}(0) = \boldsymbol{0}$, where the mass matrix is (29) and the new stiffness matrix is

$$\widetilde{\boldsymbol{S}} = \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\mathcal{A}\boldsymbol{U}(\boldsymbol{x}). \tag{48}$$

While the $m \times m$ blocks of $\boldsymbol{M}$ are obtained from the data matrices $\{\boldsymbol{D}_j\}_{j=0}^{2n-2}$ as in (31), the $m \times m$ blocks of the new stiffness matrix are computed from $\{\ddot{\boldsymbol{D}}_j\}_{j=0}^{2n-2}$, as follows

$$\begin{aligned}\widetilde{\boldsymbol{S}}_{i,j} &= \langle \boldsymbol{u}_i, \mathcal{A}\boldsymbol{u}_j\rangle = \langle \cos\left(i\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0, \mathcal{A}\cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0\rangle \\ &= \langle \boldsymbol{u}_0, \mathcal{A}\cos\left(i\tau\sqrt{\mathcal{A}}\right)\cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0\rangle \\ &= \frac{1}{2}\langle \boldsymbol{u}_0, \mathcal{A}\boldsymbol{u}_{i+j} + \mathcal{A}\boldsymbol{u}_{|i-j|}\rangle \\ &= -\frac{1}{2}\langle \boldsymbol{u}_0, \partial_t^2\boldsymbol{u}\big((i+j)\tau\big) + \partial_t^2\boldsymbol{u}\big(|i-j|\tau\big)\rangle \\ &= -\frac{1}{2}\left(\ddot{\boldsymbol{D}}_{i+j} + \ddot{\boldsymbol{D}}_{|i-j|}\right), \quad i,j = 0,\dots,n-1. \end{aligned} \tag{49}$$

We use again the block Cholesky factorization (34) of $\boldsymbol{M}$ to transform the Galerkin equation to an algebraic form that captures the causal progression of the wavefront. Multiplying (47) on the left by $\boldsymbol{R}^{-T}$ we obtain that

$$\widetilde{\boldsymbol{u}}^{\text{ROM}}(t) = \boldsymbol{R}\widetilde{\boldsymbol{g}}(t), \tag{50}$$

which takes values in $\mathbb{R}^{nm\times m}$, satisfies the algebraic analogue of the wave equation (13). Explicitly,

$$\partial_t^2\widetilde{\boldsymbol{u}}^{\text{ROM}}(t) + \boldsymbol{\mathcal{A}}^{\text{ROM}}\widetilde{\boldsymbol{u}}^{\text{ROM}}(t) = 0, \quad t > 0, \tag{51}$$

with $\widetilde{\boldsymbol{u}}^{\text{ROM}}(0) = \boldsymbol{R}\boldsymbol{e}_0 = \boldsymbol{u}_0^{\text{ROM}}$ and $\partial_t\widetilde{\boldsymbol{u}}^{\text{ROM}}(0) = \boldsymbol{0}$, where $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ is the data driven matrix

$$\boldsymbol{\mathcal{A}}^{\text{ROM}} = \boldsymbol{R}^{-T}\widetilde{\boldsymbol{S}}\boldsymbol{R}^{-1}. \tag{52}$$

Moreover, the Gram-Schmidt orthogonalization (39) and the definition (48) give that $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ can be written as the projection (45) of $\mathcal{A}$ and that the ROM wave (50) is

$$\begin{aligned}\widetilde{\boldsymbol{u}}^{\text{ROM}}(t) &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x})\widetilde{\boldsymbol{g}}(t) \\ &= \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\widetilde{\boldsymbol{u}}^{\text{GAL}}(t,\boldsymbol{x}). \end{aligned} \tag{53}$$

Comparison of the ROMs

Although both $\boldsymbol{\mathcal{P}}^{\text{ROM}}$ and $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ are given by projections on the same space $\mathbb{S}$, they capture the wave propagation in complementary ways:

The ROM propagator $\boldsymbol{\mathcal{P}}^{\text{ROM}}$ is a block tridiagonal matrix that approximates well the wave snapshots in the physical domain Ω. In fact, we can conclude from equations (27), (33), (43) and the definition (25) of $\mathbb{S}$ that the first n snapshots are captured exactly,

$$\boldsymbol{V}(\boldsymbol{x})\boldsymbol{u}_j^{\text{ROM}} = \boldsymbol{u}_j(\boldsymbol{x}), \qquad j = 0,\dots,n-1, \quad \boldsymbol{x} \in \Omega. \tag{54}$$

We also have the data match (44).

The ROM operator $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ does not capture exactly the wave snapshots, although it approximates them (see for example Herkt et al. (2013)). Data interpolation relations like (44) are also not satisfied by $\widetilde{\boldsymbol{u}}^{\text{ROM}}(t)$. However, since $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ approximates $\mathcal{A}$, which has a simple dependence on $c(\boldsymbol{x})$, it is better suited for velocity estimation. This estimation can be carried out in a layer peeling fashion, by working with data in progressively longer time windows, because the orthonormal basis in $\boldsymbol{V}(\boldsymbol{x})$ is causal. Indeed, equation (40) shows that $\boldsymbol{V}(\boldsymbol{x})\boldsymbol{e}_k$ depends only on the first k snapshots, so the ROM operator built from data at time instants $\{j\tau\}_{j=0}^{2k-2}$ senses the medium up to the depth reached by the wavefront at time $(k-1)\tau$. We refer to Appendix A for more details.

ROM based velocity estimation

We propose to estimate $c(\boldsymbol{x})$ by minimizing the misfit of the ROM operator, measured in the Frobenius norm

$$\min_{v\in\mathcal{C}} \mathcal{O}(v), \quad \mathcal{O}(v) = \|\boldsymbol{\mathcal{A}}^{\text{ROM}}(v) - \boldsymbol{\mathcal{A}}^{\text{ROM}}\|_F^2. \tag{55}$$

Here v denotes a velocity model in the search space $\mathcal{C}$, $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ is computed from the measurements (3) with Algorithm 1 given below, and $\boldsymbol{\mathcal{A}}^{\text{ROM}}(v)$ is computed with the same algorithm, from the analogue of (3) calculated by solving the wave equation in the medium with the velocity model v. The search space $\mathcal{C}$ is parametrized using some appropriate basis functions $\{\phi_l(\boldsymbol{x})\}_{l=1}^N$

$$v(\boldsymbol{x};\boldsymbol{\eta}) = c_o(\boldsymbol{x}) + \sum_{l=1}^N \eta_l\phi_l(\boldsymbol{x}), \tag{56}$$

where $c_o(\boldsymbol{x})$ is the initial guess. The optimization is then N-dimensional, for the vector $\boldsymbol{\eta} = (\eta_1,\dots,\eta_N)^T$ of coefficients in the expansion (56).

Algorithm 1 (Data-driven ROM operator)

Input: *The measurements* (3) *at time instants* $j\tau$, *for* $j = 0,\dots,2n-2$.

1. Compute $\{\boldsymbol{D}_j\}_{j=0}^{2n-2}$ *using equation* (8).

2. Compute $\{\ddot{\boldsymbol{D}}_j\}_{j=0}^{2n-2}$ *from* (8) *using, e.g., the Fourier transform, see Appendix C.*

3. Calculate $\boldsymbol{M}, \widetilde{\boldsymbol{S}} \in \mathbb{R}^{mn\times mn}$ *with the block entries*

$$\begin{aligned}\boldsymbol{M}_{i,j} &= \frac{1}{2}\big(\boldsymbol{D}_{i+j} + \boldsymbol{D}_{|i-j|}\big) \in \mathbb{R}^{m\times m}, \\ \widetilde{\boldsymbol{S}}_{i,j} &= -\frac{1}{2}\big(\ddot{\boldsymbol{D}}_{i+j} + \ddot{\boldsymbol{D}}_{|i-j|}\big) \in \mathbb{R}^{m\times m},\end{aligned}$$

for $i,j = 0,1,\dots,n-1$.

4. Perform the block Cholesky factorization $\boldsymbol{M} = \boldsymbol{R}^T\boldsymbol{R}$ *using (Druskin et al., 2018, Algorithm 5.2).*

Output: $\boldsymbol{\mathcal{A}}^{\text{ROM}} = \boldsymbol{R}^{-T}\widetilde{\boldsymbol{S}}\boldsymbol{R}^{-1}$.

To carry out the inversion in a layer stripping fashion, from the data at time instants $\{j\tau\}_{j=0}^{2k-2}$, with $k \le n$, we use the causal construction of the ROM (Appendix A) and replace $\boldsymbol{\mathcal{A}}^{\text{ROM}}(v)$ and $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ in (57) by the upper left $km \times km$ blocks of these matrices, denoted by $\big[\boldsymbol{\mathcal{A}}^{\text{ROM}}(v)\big]_k$ and $\big[\boldsymbol{\mathcal{A}}^{\text{ROM}}\big]_k$, respectively. Since $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ and thus $\big[\boldsymbol{\mathcal{A}}^{\text{ROM}}\big]_k$ are symmetric matrices, it is enough to consider their block upper triangular part in the optimization. Unlike the ROM propagator $\boldsymbol{\mathcal{P}}^{\text{ROM}}$, which is block tridiagonal, these matrices are full. However, their entries decay away from the main diagonal (see Appendix B), so we can ease the computational burden by including only the first few dm diagonals in the objective function, where d is an integer between 1 and k. For this purpose, we denote by

$$\mathrm{Rest}_{d,k} : \mathbb{R}^{km\times km} \mapsto \mathbb{R}^{dm(km-(dm-1)/2)}$$

the mapping that takes a $km \times km$ matrix, keeps only its first dm upper diagonals, including the main one, and puts their entries into a column vector, of length

$$\sum_{j=0}^{dm-1} (km - j) = dm(km - (dm - 1)/2).$$

The objective function that takes into account both the time windowing and the restriction of the ROM to a few diagonals is denoted henceforth by

$$\mathcal{O}_{d,k}(v) = \left\|\mathrm{Rest}_{d,k}\big([\boldsymbol{\mathcal{A}}^{\text{ROM}}(v) - \boldsymbol{\mathcal{A}}^{\text{ROM}}]_k \big)\right\|_2^2, \tag{57}$$

where $\|\cdot\|_2$ is the vector Euclidean norm.

Algorithm 2 (Velocity estimation)

Input: *The data driven* $\boldsymbol{\mathcal{A}}^{\text{ROM}}$.

1. Set the number of layers for the layer stripping approach to ℓ *and the number of iterations per layer to* q.

2. Choose ℓ *natural numbers* $\{k_l\}_{l=1}^{\ell}$, *satisfying*

$$1 \le k_1 \le k_2 \le \cdots \le k_\ell = n.$$

The data subset for the l^{th} *layer is* $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k_l-2}$.

3. Starting with the initial vector $\boldsymbol{\eta}^{(0)} = \boldsymbol{0}$, *proceed:*

For $l = 1, 2, \ldots, \ell$, *and* $j = 1, \ldots, q$, *set the update index* $i = (l-1)q + j$. *Compute* $\boldsymbol{\eta}^{(i)}$ *as a Gauss-Newton update for minimizing the functional*

$$\mathcal{F}_i(\boldsymbol{\eta}) = \mathcal{O}_{d,k_l}\big(v(\cdot;\boldsymbol{\eta})\big) + \mathcal{F}_i^{\text{reg}}(\boldsymbol{\eta}),$$

linearized about $\boldsymbol{\eta}^{(i-1)}$. *The term* $\mathcal{F}_i^{\text{reg}}(\boldsymbol{\eta})$ *introduces a user defined regularization penalty in the optimization.*

Output: *The velocity estimate* $c^{\text{est}}(\boldsymbol{x}) = v(\boldsymbol{x};\boldsymbol{\eta}^{(\ell q)})$.

Implementation of the inversion algorithm

In principle, the update computation at step 3 of Algorithm 2 could have a constraint on $\boldsymbol{\eta}$ to ensure that the search velocity (56) is positive. We did not need such a constraint in our numerical simulations.

There are many possible regularization penalties. For simplicity, we use the adaptive Tikhonov regularization

$$\mathcal{F}_i^{\text{reg}}(\boldsymbol{\eta}) = \mu_i \|\boldsymbol{\eta}\|_2^2, \tag{58}$$

where $\|\cdot\|_2$ is the Euclidean norm and μ_i is chosen adaptively with the following procedure: Let

$$\mathcal{R}(\boldsymbol{\eta}; d, k_l) = \mathrm{Rest}_{d,k_l}\left([\boldsymbol{\mathcal{A}}^{\text{ROM}}(v(\cdot;\boldsymbol{\eta})) - \boldsymbol{\mathcal{A}}^{\text{ROM}}]_{k_l}\right) \tag{59}$$

be the $md(2k-d+1)/2$-dimensional residual vector, whose Euclidean norm squared appears in the objective function. Its Jacobian evaluated at $\boldsymbol{\eta} = \boldsymbol{\eta}^{(i-1)}$ is the matrix

$$\boldsymbol{J}^{(i)} = \nabla_{\boldsymbol{\eta}}\mathcal{R}(\boldsymbol{\eta}^{(i-1)}; d, k_l) \in \mathbb{R}^{dm(km-(dm-1)/2)\times N}.$$

We always choose the parametrization (56) of the velocity so that the Jacobian has more rows than columns. Let $\sigma_1^{(i)} \ge \sigma_2^{(i)} \ge \cdots \ge \sigma_N^{(i)}$ be the singular values of $\boldsymbol{J}^{(i)}$. For a fixed parameter $\gamma \in (0,1)$, with smaller values corresponding to stronger regularization, we set

$$\mu_i = \big(\sigma^{(i)}_{\lfloor \gamma N\rfloor}\big)^2. \tag{60}$$

The choice of γ depends on the parametrization (56). Since it is not clear what is the resolution of the inversion, we choose to over-parametrize the velocity, and stabilize the inversion with a small γ, in the range $(0.2, 0.4)$.

The Gauss-Newton update direction for the objective function is

$$\boldsymbol{d}^{(i)} = -\Big(\big(\boldsymbol{J}^{(i)}\big)^T\boldsymbol{J}^{(i)} + \mu_i\boldsymbol{I}_N\Big)^{-1}\big(\boldsymbol{J}^{(i)}\big)^T\boldsymbol{r}^{(i)},$$

where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix and $\boldsymbol{r}^{(i)}$ is the residual vector (59) evaluated at $\boldsymbol{\eta}^{(i-1)}$. Given the update direction $\boldsymbol{d}^{(i)}$, we use a line search

$$\alpha^{(i)} = \underset{\alpha\in(0,\alpha_{\max})}{\operatorname{argmin}}\ \mathcal{F}_i\big(\boldsymbol{\eta}^{(i-1)} + \alpha\boldsymbol{d}^{(i)}\big)$$

to compute the step length $\alpha^{(i)}$, where we take $\alpha_{\max} = 3$. Then, the Gauss-Newton update is $\boldsymbol{\eta}^{(i)} = \boldsymbol{\eta}^{(i-1)} + \alpha^{(i)}\boldsymbol{d}^{(i)}$.

NUMERICAL ILLUSTRATION

In this section we give two numerical illustrations of the benefits of the velocity estimation with the ROM operator vs. FWI. The first illustration is for a two-parameter velocity model, where we can plot the objective function over the search space. The second is for the "Camembert example" introduced in (Gauthier et al. (1986)) to demonstrate the cycle skipping challenge in FWI. More numerical results are shown in the next section, where we consider noisy data.

All the results are for the source pulse

$$f(t) = \cos(\omega_o t)\exp\Big[-\frac{(2\pi B)^2t^2}{2}\Big], \tag{61}$$

with central frequency $\omega_o/(2\pi) = 6$Hz and bandwidth $B = 4$Hz. See Appendix C for details on the numerically simulated data, including the homogeneous boundary condition.

The array of m sensors is at 150m below the top boundary. The sensor spacing is 160.3m for the two-parameter velocity model and 155.5m for the Camembert example. For each simulation we specify m, the size of the rectangular domain Ω, the data sampling interval τ and the number n of snapshots that define the approximation space.

Note that to calculate τ we use $\omega_o/(2\pi) + B = 10$Hz as the essential Nyquist frequency. Thus, for $\tau = 1/(2.3 \cdot 10\text{Hz}) = 0.0435$s , the data are sampled at "2.3 points per wavelength".

Topography of the objective function

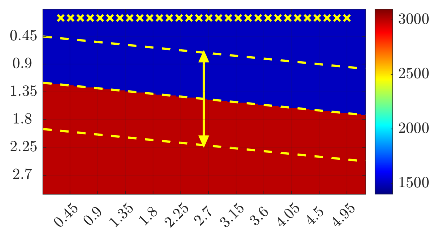

Figure 1: Velocity model used to display the topography of the objective functions. The middle dashed line shows the actual interface location, while the top and bottom dashed lines show the extent of interface location parameter sweep. All $m = 30$ sensors are shown as yellow $\times$. Distances are given in km and the velocity in m/s.

Consider the velocity model displayed in Fig. 1, in the domain $\Omega = [0, 5\text{km}] \times [0, 3\text{km}]$. It consists of two homogeneous regions separated by a slanted interface. The top region has the slower velocity $c_t = 1500$m/s, while the bottom region has the faster velocity $c_b = 3000$m/s. To plot the objective functions, we sweep a two-parameter search space $\mathcal{C}$: The first parameter is the depth of the interface in the domain $[0.47\text{km}, 1.95\text{km}]$, as measured at the leftmost point of the interface. The actual depth is at 1.2km. The second parameter is the contrast c_b/c_t in the interval $[1, 3]$. The actual contrast is 2. The angle of the interface in the search space is constant and equal to the actual angle. Note that in this exercise the search velocity is not of the form (56), as we do not run Algorithm 2. We consider a two-parameter space, so we can visualize the convexity properties of the objective functions.

In Fig. 2 we display three objective functions, calculated for $m = 30$ sensors in the array, for $n = 39$, and for the two-parameter search space $\mathcal{C}$ described above. The data sampling interval is $\tau = 0.0435$s.

The first objective function is for the FWI approach, and measures the data misfit

$$\mathcal{O}^{\text{FWI}}(v) = \sum_{k=0}^{2n-1} \left\| \text{Triu}\big(\boldsymbol{D}_k(v) - \boldsymbol{D}_k\big) \right\|_2^2, \qquad (62)$$

where $\boldsymbol{D}_k(v)$ are the $m \times m$ data matrices for the search velocity and $\text{Triu} : \mathbb{R}^{m\times m} \mapsto \mathbb{R}^{m(m+1)/2}$ is the mapping that takes a symmetric $m \times m$ matrix, extracts its upper triangular part, including the main diagonal, and arranges its entries into a $m(m+1)/2$-dimensional column vector.

The second objective function measures the misfit of the ROM propagator

$$\mathcal{O}^{\text{PROM}}(v) = \left\| \text{Triu}\big(\boldsymbol{\mathcal{P}}^{\text{ROM}}(v) - \boldsymbol{\mathcal{P}}^{\text{ROM}}\big) \right\|_2^2, \qquad (63)$$

where $\boldsymbol{\mathcal{P}}^{\text{ROM}}$ is computed from the measurements $\left\{\boldsymbol{D}_j\right\}_{j=0}^{2n-1}$ via (31)–(32) followed by (34) and (38), while $\boldsymbol{\mathcal{P}}^{\text{ROM}}(v)$ is computed from $\left\{\boldsymbol{D}_j(v)\right\}_{j=0}^{2n-1}$ using the same formulas.

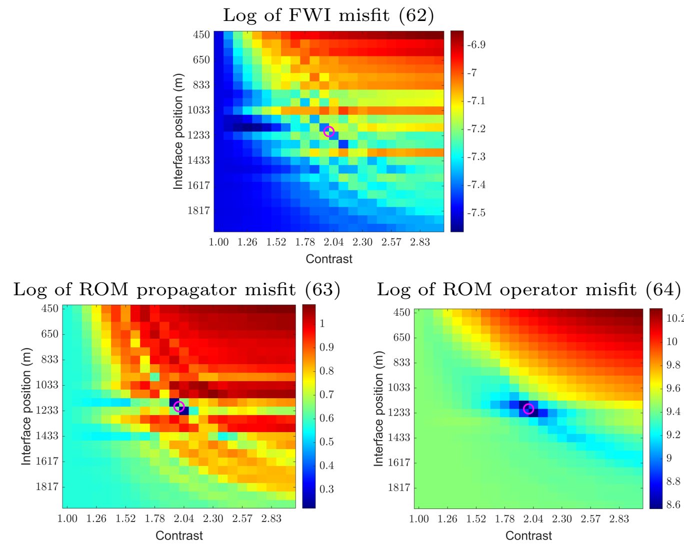

Figure 2: Logarithm of objective functions vs. the interface position and velocity contrast. The true parameters (shown in Figure 1) are indicated by $\bigcirc$.

The third objective function is what we propose in this paper, and measures the misfit of the ROM operator

$$\mathcal{O}^{\text{ROM}}(v) = \left\| \text{Triu}\big(\boldsymbol{\mathcal{A}}^{\text{ROM}}(v) - \boldsymbol{\mathcal{A}}^{\text{ROM}}\big) \right\|_2^2. \qquad (64)$$

This corresponds to the particular case $d = k = n$ of the objective function (57).

We observe in Fig. 2 that both the FWI and ROM propagator objective functions display numerous local minima, at points in the search space that are far from the true one, marked in the plots by the magenta circle. The clearly visible horizontal stripes in the plots of these objective plots are manifestations of cycle skipping. The ROM operator misfit function is smooth and has a single minimum, at the true depth and contrast. This confirms the expectation that the ROM operator misfit objective (64) and therefore (57), are superior to the FWI objective (62) for velocity estimation.

The "Camembert" example

We follow (Yang et al. (2018)) and model the "Camembert" inclusion as a disk with radius of 600m, centered at point $(1\text{km}, 1\text{km})$ in the domain $\Omega = [0, 2\text{km}] \times [0, 2.5\text{km}]$. The setup is illustrated in Fig. 3, where $c(\boldsymbol{x})$ equals 4000m/s in the inclusion and 3000m/s outside. The data sampling interval is $\tau = 0.0435$s, the number of sensors is $m = 10$ and $n = 16$.

The search space $\mathcal{C}$ has dimension $N = 20 \times 20 = 400$, and the velocity is parametrized as in equation (56), with the constant initial guess $c_o(\boldsymbol{x}) = 3000$m/s and the Gaussian basis functions

$$\phi_l(\boldsymbol{x}) = \frac{1}{2\pi\sigma_\phi\sigma_\phi^\perp} \exp\Big[-\frac{(x^\perp - x_l^\perp)^2}{2(\sigma_\phi^\perp)^2} - \frac{(x - x_l)^2}{2\sigma_\phi^2} \Big], \qquad (65)$$

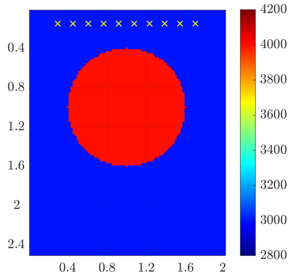

Figure 3: "Camembert" velocity model. All $m = 30$ sensors are shown as yellow $\times$. The axes are in km and the velocity shown in the colorbar is in m/s.

with standard deviation $\sigma_\phi^\perp = 55.5$m in the horizontal (cross-range) direction and $\sigma_\phi = 69.4$m in the range vertical (range) direction. Here we use the system of coordinates $\boldsymbol{x} = (x^\perp, x)$, with cross-range $x^\perp$ and range x. The centers of the Gaussians are at the locations $\boldsymbol{x}_l = (x_l^\perp, x_l)$ on a uniform 20×20 grid that discretizes the imaging domain $\Omega_{\rm im} = [95\text{m}, 1905\text{m}] \times [119\text{m}, 2381\text{mm}] \subset \Omega$. Note that $2\sigma_\phi$ and $2\sigma_\phi^\perp$ are smaller than half the wavelength $c_o/(10\text{Hz}) = 300$m corresponding to the essential Nyquist frequency. Hence, the problem is over-parametrized and we stabilize the inversion with the adaptive Tikhonov regularization (58) with $\gamma = 0.25$.

We show in the left column of Fig. 4 the results of the inversion obtained with Algorithm 2, implemented with $\ell = 9$, the number of iterations per layer $q = 4$, and with the restriction parameter $d = n$.

The plots in the right column of Fig. 4 are for the FWI approach, with the same time windowing of the data as in the ROM based inversion. Here we minimize the objective function

$$\mathcal{F}_i^{\text{FWI}}(\boldsymbol{\eta}) = \mathcal{O}^{\text{FWI}}\big(v(\cdot;\boldsymbol{\eta})\big) + \mu_i^{\text{FWI}} \|\boldsymbol{\eta}\|_2^2, \tag{66}$$

with the Tikhonov regularization parameter μ_i^{FWI} computed with a similar procedure to that used for the ROM inversion: If we let $\mathcal{R}^{\text{FWI}}(\boldsymbol{\eta}) \in \mathbb{R}^{nm(m+1)}$ be the residual vector, with entries

$$\big(\mathcal{R}_j^{\text{FWI}}(\boldsymbol{\eta})\big)_{j=km(m+1)/2+1}^{(k+1)m(m+1)/2} = \text{Triu}\big(\boldsymbol{D}_k(v) - \boldsymbol{D}_k\big),$$

for $k = 0, \ldots, 2n-1$, its Jacobian evaluated at $\boldsymbol{\eta} = \boldsymbol{\eta}^{(i-1)}$ is the $nm(m+1) \times N$ matrix $\boldsymbol{J}^{\text{FWI},(i)} = \nabla_{\boldsymbol{\eta}} \mathcal{R}^{\text{FWI}}(\boldsymbol{\eta}^{(i-1)})$. We assume $N \leq nm(m+1)$. Then, for the same fixed parameter γ used in Algorithm 2, we set $\mu_i^{\text{FWI}} = \big(\sigma_{\lfloor \gamma N \rfloor}^{\text{FWI},(i)}\big)^2$, where $\big\{\sigma_j^{\text{FWI},(i)}\big\}_{j=1}^N$ are the singular values of $\boldsymbol{J}^{\text{FWI},(i)}$, sorted in decreasing order.

The results in Fig. 4 show that the ROM inversion gives a much better estimate of $c(\boldsymbol{x})$. This estimate improves as we iterate, and by the time we reach the 60^{th} step, the circular inclusion is reconstructed well. The FWI approach does not improve much after the 10^{th} step, indicating that the optimization is stuck in a local minimum. While the top and arguably the bottom of the inclusion are correctly

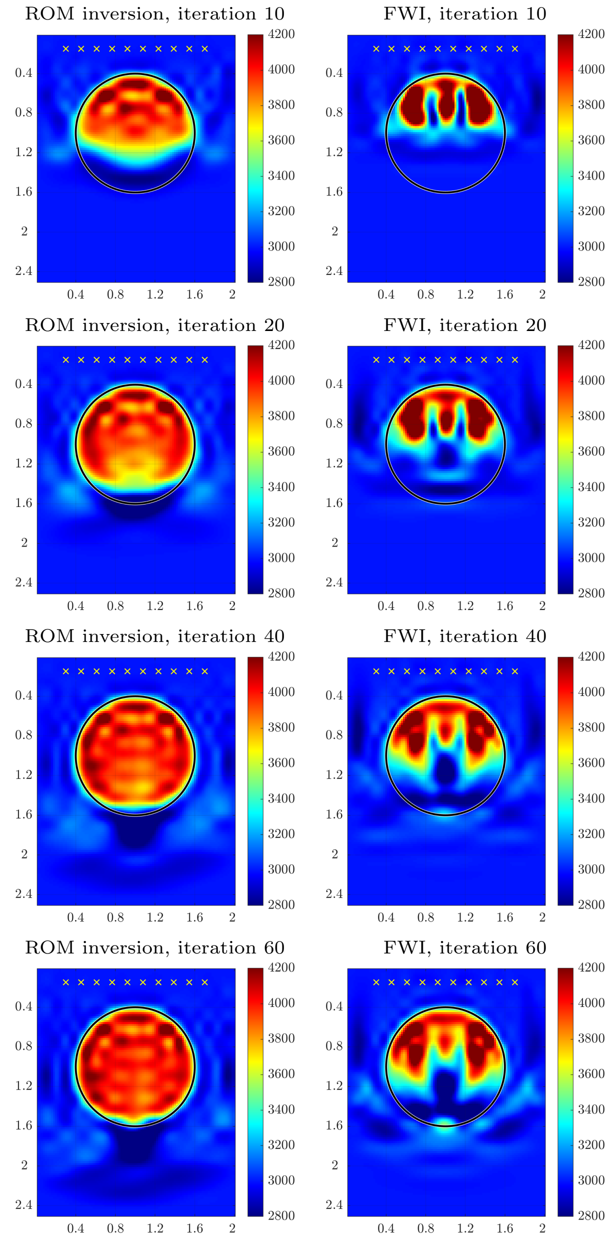

Figure 4: Estimated velocity after $10 - 60$ Gauss-Newton iterations. The left column is for the ROM operator based inversion. The right column is for FWI. The true inclusion boundary is shown as a black circle. The axes, color scale and units in the colorbar are as in Fig. 3.

located, FWI fails to fill in the inclusion with the correct velocity, overestimating it in the upper half of the disk and underestimating it in the lower half.

INVERSION WITH NOISY DATA

In this section we discuss how to mitigate noise effects in the ROM based velocity estimation. Algorithm 2 has a regularization penalty on the search velocity built into it, but when the data are noisy, we also need the regularize the construction of $\mathcal{A}^{\text{ROM}}$. This is not a straightforward modification of Algorithm 1, as the success of the inversion depends on the ROM having an algebraic structure that captures the causal propagation of the wavefront inside the medium. We explain the ROM regularization next, and then show inversion results with noisy data.

Regularization of the ROM operator

Let us denote by $\{\boldsymbol{D}_j^{\text{N}}\}_{j=0}^{2n-1}$ the data matrices contaminated with noise (see Appendix C). These matrices should be symmetric, due to the source-receiver reciprocity, but the noisy matrices are not symmetric, so we first symmetrize them by transforming $\boldsymbol{D}_j^{\text{N}}$ into $\frac{1}{2}(\boldsymbol{D}_j^{\text{N}} + \boldsymbol{D}_j^{\text{N}^T})$.

The mass and stiffness matrices computed from the noisy data as in step 3 of Algorithm 1, are denoted by $\boldsymbol{M}^{\text{N}}$ and $\widetilde{\boldsymbol{S}}^{\text{N}}$. In theory, they should be positive definite matrices, but due to noise they may have a number of eigenvalues that are negative or zero. This is critical in the case of $\boldsymbol{M}^{\text{N}}$, because we need the inverse of its block Cholesky square root to compute $\mathcal{A}^{\text{ROM}}$.

A natural way of regularizing $\boldsymbol{M}^{\text{N}}$ is via projection on the space spanned by the leading eigenvectors. Thus, let

$$\boldsymbol{M}^{\text{N}} = \boldsymbol{Z}^{\text{N}}\boldsymbol{\Lambda}^{\text{N}}(\boldsymbol{Z}^{\text{N}})^T, \tag{67}$$

be the eigendecomposition of $\boldsymbol{M}^{\text{N}}$, where $\boldsymbol{Z}^{\text{N}}$ is the orthogonal matrix of eigenvectors and $\boldsymbol{\Lambda}^{\text{N}} = \text{diag}(\lambda_1^{\text{N}}, \ldots, \lambda_{nm}^{\text{N}})$ is the diagonal matrix of eigenvalues, in descending order. We wish to keep the eigenvalues that are larger than the noise contribution (see Appendix D). Since we work with $m\times m$ blocks, we choose the cut-off at index rn, for integer r satisfying $1 \le r < n$, and use the first rm eigenvectors, stored in

$$\boldsymbol{Z}^{\text{N},r} = (Z_{jl}^{\text{N}})_{1\le j\le nm, 1\le l\le rm} \in \mathbb{R}^{nm\times rm} \tag{68}$$

to define the projected mass matrix

$$\boldsymbol{\Lambda}^{\text{N},r} = (\boldsymbol{Z}^{\text{N},r})^T\boldsymbol{M}^{\text{N}}\boldsymbol{Z}^{\text{N},r} = \text{diag}\big(\lambda_1^{\text{N}}, \ldots, \lambda_{rm}^{\text{N}}\big). \tag{69}$$

The resulting $\boldsymbol{\Lambda}^{\text{N},r}$ is well-conditioned, but it does not have the block Hankel + Toeplitz structure deduced from the causal propagation of the wave (recall equation (31)). Thus, we need an additional transformation to recover the algebraic causal structure of the regularized mass matrix. The desired transformation cannot be obtained by looking at the ROM operator construction alone, because all we know about the algebraic structure of $\mathcal{A}^{\text{ROM}}$ is that its entries decay away from the main diagonal. However, we can get the transformation using the ROM propagator, because of the following facts:

1. Both the ROM propagator $\mathcal{P}^{\text{ROM}}$ and the ROM operator $\mathcal{A}^{\text{ROM}}$ are built from the same mass matrix, so the projection (69) affects both these ROMs.

2. We have the time stepping scheme (37) where $\mathcal{P}^{\text{ROM}}$ propagates the snapshots in the ROM space from one time instant to the next.

3. Algebraic causality means that the first n snapshots in the ROM space form a block upper triangular matrix. From (37) we conclude that this is equivalent to saying that $\mathcal{P}^{\text{ROM}}$ is block tridiagonal, because by construction $\boldsymbol{u}_0^{\text{ROM}}$ is an $nm\times m$ matrix with all but the first m rows equal to zero.

Thus, we proceed as follows: First, we compute the ROM propagator stiffness matrix $\boldsymbol{S}^{\text{N}}$, with blocks given in terms of the noisy data $\big\{\boldsymbol{D}_j^{\text{N}}\big\}_{j=0}^{2n-1}$ as in equation (32). Then, we project this matrix onto the range of $\boldsymbol{Z}^{\text{N},r}$:

$$\boldsymbol{S}^{\text{N},r} = (\boldsymbol{Z}^{\text{N},r})^T\boldsymbol{S}^{\text{N}}\boldsymbol{Z}^{\text{N},r}, \tag{70}$$

and we compute the analogue of the operator ROM (38)

$$\boldsymbol{P}^{\text{N},r} = (\boldsymbol{\Lambda}^{\text{N},r})^{-1/2}\boldsymbol{S}^{\text{N},r}(\boldsymbol{\Lambda}^{\text{N},r})^{-1/2}. \tag{71}$$

This is a symmetric, positive definite matrix that can be put in block tridiagonal form using the block-Lanczos algorithm (Golub and Van Loan, 2013, Chapter 10) on $\boldsymbol{P}^{\text{N},r}$, with starting $rm\times m$ matrix $(\boldsymbol{\Lambda}^{\text{N},r})^{-1/2}(\boldsymbol{Z}^{\text{N},r})^T\boldsymbol{e}_0$. This generates an orthogonal matrix $\boldsymbol{Q}^{\text{N},r} \in \mathbb{R}^{rm\times rm}$ such that

$$\mathcal{P}^{\text{ROM},r} = (\boldsymbol{Q}^{\text{N},r})^T\boldsymbol{P}^{\text{N},r}\boldsymbol{Q}^{\text{N},r} \tag{72}$$

is an $rm\times rm$ block tridiagonal matrix, which we call the regularized ROM propagator.

The matrix $\mathcal{P}^{\text{ROM},r}$ itself is irrelevant for our velocity estimation approach. It is the orthogonal transformation given by $\boldsymbol{Q}^{\text{N},r}$ that we need, which restores the desired algebraic causality of the regularized mass matrix. Using this transformation we can obtain the noisy ROM operator with the following procedure: Compute the block Cholesky factorization of the transformed mass matrix

$$(\boldsymbol{Q}^{\text{N},r})^T\boldsymbol{\Lambda}^{\text{N},r}\boldsymbol{Q}^{\text{N},r} = (\Pi^{\text{N},r})^T\boldsymbol{M}^{\text{N}}\Pi^{\text{N},r} = (\boldsymbol{R}^{\text{N},r})^T\boldsymbol{R}^{\text{N},r}, \tag{73}$$

where

$$\Pi^{\text{N},r} = \boldsymbol{Z}^{\text{N},r}\boldsymbol{Q}^{\text{N},r}. \tag{74}$$

and $\boldsymbol{R}^{\text{N},r} \in \mathbb{R}^{rm\times rm}$ is block upper triangular and well conditioned, due to the spectral truncation in (69). Then, using the data driven stiffness matrix $\widetilde{\boldsymbol{S}}^{\text{N}}$ we obtain the regularized operator ROM as

$$\mathcal{A}^{\text{ROM},r} = (\boldsymbol{R}^{\text{N},r})^{-T}(\Pi^{\text{N},r})^T\widetilde{\boldsymbol{S}}^{\text{N}}\Pi^{\text{N},r}(\boldsymbol{R}^{\text{N},r})^{-1}. \tag{75}$$

Equation (75) gives the regularization of the data driven ROM operator construction. For the inversion, we also need the ROM operator for the search velocity $v(\boldsymbol{x};\boldsymbol{\eta})$ computed via the same chain of transformations, using the same matrix (74): Let $\boldsymbol{M}(v)$ and $\widetilde{\boldsymbol{S}}(v)$ be the mass and stiffness matrices calculated as in step 3 of Algorithm 1 from the data computed numerically in the medium with velocity $v(\boldsymbol{x},\boldsymbol{\eta})$. We compute the block Cholesky factorization

$$(\Pi^{\text{N},r})^T\boldsymbol{M}(v)\Pi^{\text{N},r} = \boldsymbol{R}^r(v)^T\boldsymbol{R}^r(v), \tag{76}$$

where r is an index (not a power). Then, the ROM operator at the search velocity v is given by

$$\mathcal{A}^{\text{ROM},r}(v) = \boldsymbol{R}^r(v)^{-T}(\Pi^{\text{N},r})^T\widetilde{\boldsymbol{S}}(v)\Pi^{\text{N},r}\boldsymbol{R}^r(v)^{-1}. \tag{77}$$

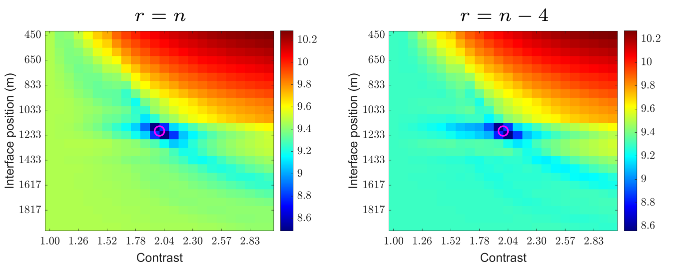

Figure 5: Logarithm of objective function (78) vs. the interface position and velocity contrast. The true parameters (shown in Fig. 1) are indicated by ◯.

The velocity inversion is carried out as in Algorithm 2, with $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ and $\boldsymbol{\mathcal{A}}^{\text{ROM}}(v)$ in (57) replaced by the regularized $\boldsymbol{\mathcal{A}}^{\text{ROM},r}$ and $\boldsymbol{\mathcal{A}}^{\text{ROM},r}(v)$, respectively.

Note that due to the block algebra, even if we do not use a spectral truncation, i.e., set $r = n$, the ROM operator (76) is not identical to the one computed with Algorithm 1. Nevertheless, they behave the same with respect to the inversion, as illustrated in Fig. 5, where we plot the logarithm of the objective function

$$\mathcal{O}^{\text{ROM},r}(v) = \left\| \text{Triu}\big(\boldsymbol{\mathcal{A}}^{\text{ROM},r}(v) - \boldsymbol{\mathcal{A}}^{\text{ROM},r}\big) \right\|_2^2 \qquad (78)$$

for the same experiment as in Figs. 1 and 2, for the cases $r = n$ and $r = n - 4$. There is little difference between the bottom right plot in Fig. 2 and the left plot in Fig. 5.

Numerical results: Marmousi model

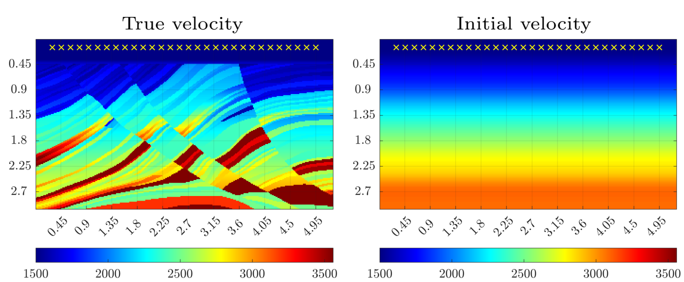

Figure 6: The section of the Marmousi model (left) and the initial guess $c_o(\boldsymbol{x})$ (right). All $m = 30$ sensors are shown as yellow ×. The axes are in km and the units in the colorbar are in m/s.

We now present the velocity estimation results for a section of the Marmousi model shown in Fig. 6, where we exclude the portion of the water down to depth 266m. The domain is $\Omega = [0, 5.25\text{km}] \times [0, 3\text{km}]$. There are $m = 30$ sensors located at depth $150m$ and they emit the same pulse (61). The data sampling for the ROM construction is $\tau = 0.0435$s and the number of snapshots that span the approximation space is $n = 40$.

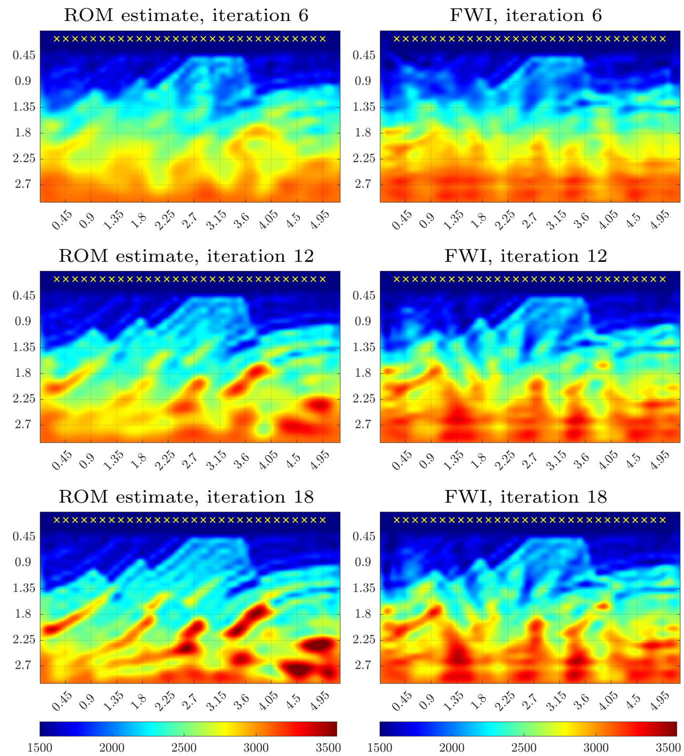

Figure 7: Velocity estimates for the Marmousi model after $6, 12$ and 18 Gauss-Newton iterations. Left column: ROM based approach. Right column: FWI approach. The axes and colorbar are as in Fig. 6.

In the left plots of Fig. 7 we show the ROM based inversion results obtained from data contaminated with 1% additive noise described in Appendix C. We used $\ell = 6$ layers in Algorithm 2, with $q = 3$ iterations per layer, and the restriction parameter $d = 10$. The ROM operator is regularized as explained above and the spectral threshold parameter is set at $r = n - 9 = 31$. The velocity is parametrized as in equation (56), with the initial guess $c_o(\boldsymbol{x})$ displayed in the right plot of Fig. 6. We used $N = 50 \times 30 = 1500$ Gaussian basis functions (65), with standard deviations $\sigma_\phi^\perp = 60$m, and $\sigma_\phi = 56.4$m. The peaks of the Gaussians are on a uniform 50×30 grid discretizing the imaging domain $\Omega_{\text{im}} = [103\text{m}, 5147\text{m}] \times [97\text{m}, 2903\text{m}] \subset \Omega$.

The right plots in Fig. 7 show the FWI results computed for noiseless data. We use the same parametrization of the search velocity and invert in $\ell = 6$ layers with the same data windowing as in the ROM based inversion.

We note that the ROM based inversion captures correctly many features of the Marmousi model, and continues to improve with the iterations. The imaging near the bottom boundary can be improved further by extending the depth of the domain Ω. Because in the ROM construction we need the wavefront to progress downward at

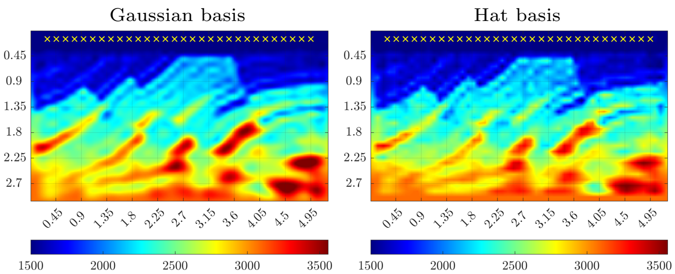

Figure 8: Comparison of the ROM based velocity estimates for the Marmousi model, after 18 Gauss-Newton iterations, for the velocity parametrized with Gaussian basis functions (left) and hat functions (right). The axes and colorbar are as in Fig. 6.

all times[‡], we limit the recording of the data to less than the estimated round trip travel time from the sensors to the bottom boundary, so we lose resolution there.

We also note that the FWI approach recovers the top features of the Marmousi model. However, the velocity estimate does not improve much after the 12$^{\text{th}}$ iteration and the result is far from the true model. Effectively, FWI is stuck in a local minimum.

In Fig. 8 we give a side-by-side comparison of the ROM based inversion result obtained with two different choices of the basis functions in the parametrization (56) of the search velocity. The Gaussian ones given in (65) and the commonly used piecewise linear hat functions, which interpolate between the values of 0 and 1 on the same 50×30 inversion grid. The result with the Gaussian basis looks smoother, as expected, but the point of the plot is to illustrate that the inversion is not sensitive to the parametrization of the search velocity, once the inversion grid is fixed.

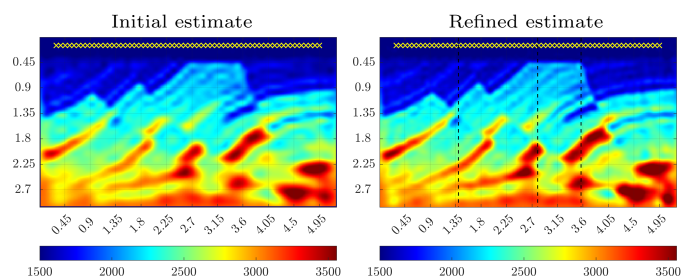

Figure 9: Velocity estimate for the Marmousi model obtained from data gathered with more sensors and at smaller time interval τ (right plot). The initial estimate (left plot) is the ROM estimate in the bottom left plot of Fig. 8. The axes and colorbar are as in Fig. 6.

We show in Fig. 9 how the velocity estimation improves if we double the number of sensors ($m = 60$), decrease the time sampling to $\tau = 0.0333$s and increase n to 50, while also setting $r = n - 17 = 33$. The inversion is carried out as above, except that the parametrization of the velocity is with $N = 75 \times 38 = 2850$ Gaussian functions with $\sigma_\phi^\perp = 40.2$m, and $\sigma_\phi = 44.8$m. The initial guess of the velocity is shown in the left plot, and it corresponds to ROM estimate shown in Fig 7. Since the initial velocity estimate is already very good, it is sufficient to perform $q = 4$ Gauss-Newton iterations for a single layer $\ell = 1$ using all the available data, i.e., $k_1 = r$. We note that the resulting refined velocity estimate sharpens the boundaries of the features and improves their contrast.

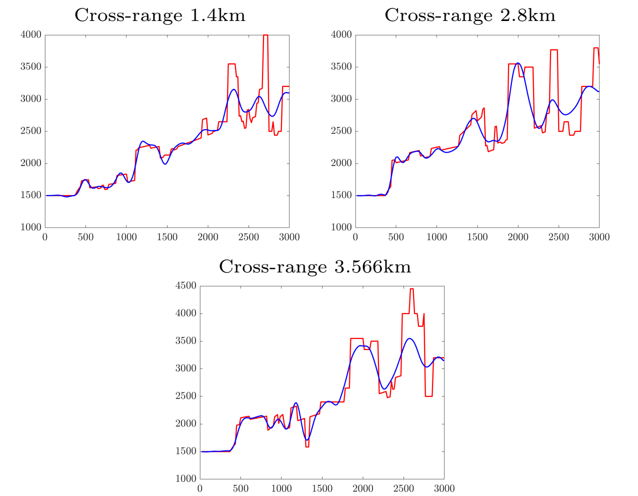

Figure 10: Vertical slices of the true velocity (red lines) and refined ROM estimate (blue lines) at cross-ranges shown as dashed lines in the right plot in Fig. 9. The abscissa is in m and the ordinate in m/s.

Finally, to illustrate better the quality of the refined ROM estimate in the 225 of Fig. 9, we display in Fig. 10 the true and reconstructed velocity for three vertical slices, at cross-ranges 1.4km, 2.8km and 3.566km. We note again that the reconstruction is accurate away from the bottom boundary, where the results can be improved by extending the depth of the domain Ω and the recording time, as explained above.

SUMMARY

We introduced a novel approach for velocity estimation based on a reduced order model (ROM) of the wave operator. The ROM is computed from the data gathered by an array of sensors that play the dual role of sources and receivers. No prior information of the medium is used, except for the assumption that the velocity is known in the immediate vicinity of the sensors. While the mapping from the data to the ROM is nonlinear, we can compute it using efficient numerical linear algebra algorithms. We explain that the ROM is an approximation of the wave operator on a space defined by the snapshots of the wave field at uniformly spaced time instants. This space is not

[‡]If this does not hold, the snapshots that define the approximation space $\mathbb{S}$ may be linearly dependent.

known and neither is the wave operator. Yet, we can compute its approximation, the ROM, from the data. We describe the properties of the ROM and formulate a velocity estimation algorithm that minimizes the ROM misfit. We also explain how to regularize the ROM in order to mitigate additive noise. We demonstrate with numerical simulations that the ROM misfit objective function is better than the nonlinear least squares data misfit used in full waveform inversion (FWI). In particular, for a low-dimensional velocity model where we can plot the objective functions, we obtain that the ROM misfit objective function is convex, while the FWI objective function displays multiple local minima. We present velocity estimation results for two well known models where FWI is known to fail in the absence of an excellent initial guess: the "Camembert" model and the "Marmousi" model.

ACKNOWLEDGMENTS

This material is based upon research supported in part by the U.S. Office of Naval Research under award number N00014-21-1-2370 to Borcea and Mamonov. Borcea, Garnier and Zimmerling also acknowledge support from the AFOSR awards FA9550-21-1-0166 and FA9550-22-1-0077. Zimmerling also acknowledges support from the National Science Foundation under Grant No. 2110265.

APPENDIX A
CAUSAL CONSTRUCTION OF THE ROM

Here we prove that the upper left $km \times km$ block of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$, denoted by $\big[\boldsymbol{\mathcal{A}}^{\text{ROM}}\big]_k$, is the ROM operator computed by Algorithm 1 from the data subset $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k-2}$, for any $k = 1, \ldots, n$.

Let us begin by writing $\big[\boldsymbol{\mathcal{A}}^{\text{ROM}}\big]_k$ from equation (52)

$$\begin{aligned}\big[\boldsymbol{\mathcal{A}}^{\text{ROM}}\big]_k &= \big(\boldsymbol{I}_{km} \quad \boldsymbol{0}\big)\, \boldsymbol{R}^{-T} \widetilde{\boldsymbol{S}} \boldsymbol{R}^{-1} \begin{pmatrix}\boldsymbol{I}_{km}\\ \boldsymbol{0}\end{pmatrix} \\ &= \Big(\big[\boldsymbol{R}\big]_k^{-T} \quad \boldsymbol{0}\Big)\, \widetilde{\boldsymbol{S}} \begin{pmatrix}\big[\boldsymbol{R}\big]_k^{-1}\\ \boldsymbol{0}\end{pmatrix} \\ &= \big[\boldsymbol{R}\big]_k^{-T} \big[\widetilde{\boldsymbol{S}}\big]_k \big[\boldsymbol{R}\big]_k^{-1} \end{aligned} \tag{A-1}$$

where $\boldsymbol{I}_{km}$ is the $km \times km$ identity matrix and $\big[\widetilde{\boldsymbol{S}}\big]_k$ and $\big[\boldsymbol{R}\big]_k$ are the upper left $km \times km$ blocks of $\widetilde{\boldsymbol{S}}$ and $\boldsymbol{R}$, respectively. Here we used that $\boldsymbol{R}$ is block upper triangular, and so is its inverse. Moreover, the upper left $km \times km$ block of $\boldsymbol{R}^{-1}$ is the same as the inverse of $\big[\boldsymbol{R}\big]_k$.

At step 3, Algorithm 1 computes from $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k-2}$ the upper left $km \times km$ block of $\boldsymbol{M}$, denoted by $\big[\boldsymbol{M}\big]_k$, and also $\big[\widetilde{\boldsymbol{S}}\big]_k$. From the Cholesky factorization (34) and the block upper triangular structure of $\boldsymbol{R}$ we get

$$\big[\boldsymbol{M}\big]_k = \big(\boldsymbol{I}_{km} \quad \boldsymbol{0}\big)\, \boldsymbol{R}^T \boldsymbol{R} \begin{pmatrix}\boldsymbol{I}_{km}\\ \boldsymbol{0}\end{pmatrix} = \big[\boldsymbol{R}\big]_k^T \big[\boldsymbol{R}\big]_k. \tag{A-2}$$

Thus, $\big[\boldsymbol{R}\big]_k$ is the Cholesky square root of $\big[\boldsymbol{M}\big]_k$, computed in Algorithm 1, and the result follows from (A-1).

APPENDIX B
ALGEBRAIC STRUCTURE OF THE ROM

We explain here that the entries of the ROM operator $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ decay away from the main diagonal, which is why we can use the restriction mapping $\text{Rest}_{d,k}$ to reduce the computational cost of inversion. Let us write

$$\boldsymbol{V}(\boldsymbol{x}) = \big(\boldsymbol{v}_0(\boldsymbol{x}), \ldots, \boldsymbol{v}_{n-1}(\boldsymbol{x})\big),$$

where $\boldsymbol{v}_j(\boldsymbol{x}) \in \mathbb{R}^{1\times m}$, for $j = 0, \ldots, n-1$. We obtain from (45) that the $m \times m$ blocks of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ are

$$\boldsymbol{\mathcal{A}}^{\text{ROM}}_{i,j} = \int_\Omega d\boldsymbol{x}\, \boldsymbol{v}_i^T(\boldsymbol{x}) \mathcal{A} \boldsymbol{v}_j(x), \qquad i,j = 0, \ldots, n-1. \tag{B-1}$$

Moreover, the Gram-Schmidt orthogonalization (39) gives $\boldsymbol{u}_j(\boldsymbol{x}) = \sum_{q=0}^{j} \boldsymbol{v}_q(\boldsymbol{x}) \boldsymbol{R}_{q,j}$, and conversely

$$\boldsymbol{v}_j(\boldsymbol{x}) = \sum_{q=0}^{j} \boldsymbol{u}_q(\boldsymbol{x}) \boldsymbol{\Gamma}_{q,j}, \tag{B-2}$$

where

$$\boldsymbol{\Gamma} = \boldsymbol{R}^{-1} = \begin{pmatrix} \boldsymbol{\Gamma}_{0,0} & \boldsymbol{\Gamma}_{0,1} & \ldots & \boldsymbol{\Gamma}_{0,n-1} \\ \boldsymbol{0} & \boldsymbol{\Gamma}_{1,1} & \ldots & \boldsymbol{\Gamma}_{1,n-1} \\ \vdots & \vdots & \vdots & \boldsymbol{\Gamma}_{n-1,n-1} \end{pmatrix}. \tag{B-3}$$

Now let us substitute (B-2) into (B-1), to obtain

$$\begin{aligned}\boldsymbol{\mathcal{A}}^{\text{ROM}}_{i,j} &= \sum_{q=0}^{j} \int_\Omega d\boldsymbol{x}\, \boldsymbol{v}_i^T(\boldsymbol{x}) \mathcal{A} \boldsymbol{u}_q(\boldsymbol{x}) \boldsymbol{\Gamma}_{q,j} \\ &= -\sum_{q=0}^{j} \int_\Omega d\boldsymbol{x}\, \boldsymbol{v}_i^T(\boldsymbol{x}) \partial_t^2 \boldsymbol{u}(q\tau, \boldsymbol{x}) \boldsymbol{\Gamma}_{q,j}, \end{aligned} \tag{B-4}$$

where the last equality is due to the wave equation (13). We use next the Whittaker-Shannon interpolation formula, which says that if τ satisfies the Nyquist criterion,

$$u(t, \boldsymbol{x}) = \sum_{s=-\infty}^{\infty} u_{|s|}(\boldsymbol{x}) \text{sinc}\Big[\frac{\pi(t - s\tau)}{\tau}\Big]. \tag{B-5}$$

Differentiating twice and evaluating at $t = q\tau$ we get

$$\tau^2 \partial_t^2 u(q\tau, \boldsymbol{x}) = \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} u_{|q-s|}(\boldsymbol{x}) - \frac{\pi^2}{3} u_q(\boldsymbol{x}),$$

and substituting into (B-4) we obtain

$$\begin{aligned}\boldsymbol{\mathcal{A}}^{\text{ROM}}_{i,j} &= \frac{1}{\tau^2} \sum_{q=0}^{j} \boldsymbol{\Gamma}_{q,j} \Big\{ \int_\Omega d\boldsymbol{x}\, \boldsymbol{v}_i^T(\boldsymbol{x}) \boldsymbol{u}_q(\boldsymbol{x}) \\ &\quad - \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} \int_\Omega d\boldsymbol{x}\, \boldsymbol{v}_i^T(\boldsymbol{x}) u_{|q-s|}(\boldsymbol{x}) \Big\} \\ &= \frac{1}{\tau^2} \sum_{q=0}^{j} \boldsymbol{\Gamma}_{q,j} \Big\{ \boldsymbol{R}_{i,q} - \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} \boldsymbol{R}_{i,|q-s|} \Big\}. \end{aligned} \tag{B-6}$$

To avoid boundary terms, we have assumed in this formula a large n so we can take $n \to \infty$.

Since $\boldsymbol{\Gamma}_{q,j} = 0$ for $q > j$, and $\boldsymbol{R}_{i,q} = 0$ for $i > q$, the first term on the right-hand side of (B-6) is zero for $i > j$. But we are interested only in the block upper triangular part of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ (i.e., $i \le j$), due to symmetry, so this first term contributes only to the main block diagonal. The other block diagonals are due to the series in (B-6). Each term in this series adds an s^{th} diagonal, whose entries decay as $1/s^2$. Thus, only the first few block diagonals are large.

APPENDIX C
NUMERICALLY SIMULATED DATA

The data for the numerical experiments are computed with a time-domain wave equation solver for (13)–(14), with $\mathcal{A}$ discretized on a uniform grid with a five point finite difference stencil. We use homogeneous Dirichlet boundary conditions at $\partial\Omega$. The second time derivative is approximated by a three point finite difference scheme, on a fine time grid with step $\tau_{\text{f}} = \tau/20$. We get the finely sampled data $\boldsymbol{D}_k^{\text{f}}$, for $k = 0, 1, \ldots, n_{\text{f}}$, where $n_{\text{f}} = 20(2n-1)$.

The noisy data are computed as follows: Define

$$\beta = \frac{b}{m\sqrt{n_{\text{f}}+1}} \left(\sum_{k=0}^{n_{\text{f}}} \|\boldsymbol{D}_k^{\text{f}}\|_F^2 \right)^{1/2}, \tag{C-1}$$

where b is the desired noise level, e.g., $b = 10^{-2}$ for 1% noise. Then, the contaminated finely sampled data is obtained by adding to $\boldsymbol{D}_k^{\text{f}}$ a realization of an $m \times m$ random matrix with independent, normally distributed entries with mean zero and standard deviation β for each $k = 1, \ldots, n_{\text{f}}$. Since the data at time zero is computed in the known medium near the sensors, we exclude $k = 0$. To simplify notation, hereafter we denote by $\boldsymbol{D}_k^{\text{f}}$ both the noiseless and the noise contaminated case.

We now explain how we compute the second derivative data matrices. We begin by extending the finely sampled data evenly in discrete time to get $\boldsymbol{D}_j^{\text{fe}}$, $j = -n_{\text{f}}, \ldots, n_{\text{f}}$, with $\boldsymbol{D}_k^{\text{f}} = \boldsymbol{D}_{\pm k}^{\text{fe}}$, $k = 0, 1, \ldots, n_{\text{f}}$. Then, we take the discrete Fourier transform of $\left(\boldsymbol{D}_j^{\text{fe}}\right)_{j=-n_{\text{f}}}^{n_{\text{f}}}$ and differentiate in the Fourier domain after using a sharp cutoff low-pass filter intended to stabilize the calculation. The cutoff frequency is at $\omega_o/(2\pi) + 4B = 22$Hz. We take the inverse Fourier transform to obtain $\ddot{\boldsymbol{D}}_j^{\text{fe}}$, at $j = -n_{\text{f}}, \ldots, n_{\text{f}}$, the finely sampled second derivative data. Finally, we subsample both $\boldsymbol{D}_j^{\text{fe}}$ and $\ddot{\boldsymbol{D}}_j^{\text{fe}}$ to get

$$\boldsymbol{D}_k = \boldsymbol{D}_{20k}^{\text{fe}}, \quad \ddot{\boldsymbol{D}}_k = \ddot{\boldsymbol{D}}_{20k}^{\text{fe}}, \quad k = 0, 1, \ldots, 2n-1. \tag{C-2}$$

APPENDIX D
CHOICE OF THE REGULARIZATION THRESHOLD

Here we explain how we choose the regularization threshold r for the ROM regularization procedure (68)–(77). The idea is that r can be determined from the part of

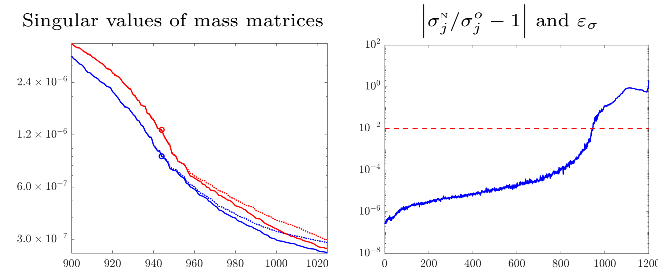

Figure D.1: Regularization threshold illustration. Left plot shows the singular values of mass matrices: $\boldsymbol{M}$ (solid red), $\boldsymbol{M}^{\text{N}}$ (dotted red), $\boldsymbol{M}(c_o)$ (solid blue) and $\boldsymbol{M}^{\text{N}}(c_o)$ (dotted blue); the circles correspond to $j = R^{\text{N}}$. Right plot: left-hand side of (D-2) (solid blue) and ε_σ (dashed red). The abscissa is the singular value index j.

the spectrum of the noisy mass matrix $\boldsymbol{M}^{\text{N}}$ that is perturbed little by the noise. This can be estimated using the background mass matrix $\boldsymbol{M}(c_o)$ corresponding to the initial guess velocity $c_o(\boldsymbol{x})$, and an estimation of the additive noise. For the latter, we make the key observation that the matrices

$$\boldsymbol{E}_j^{\text{N}} = \frac{1}{\sqrt{2}} \left(\boldsymbol{D}_j^{\text{N}} - (\boldsymbol{D}_j^{\text{N}})^T \right), \quad j = 0, \ldots, 2n-1, \tag{D-1}$$

can be considered as realizations of the additive noise. Indeed, the true wave signals are reciprocal, i.e., $\boldsymbol{D}_j$ are symmetric matrices, while the additive noise is not.

Consider the mass matrices $\boldsymbol{M}(c_o)$ and $\boldsymbol{M}^{\text{N}}(c_o)$ computed by Algorithm 1 from the noiseless background data $\{\boldsymbol{D}_j(c_o)\}_{j=0}^{2n-1}$ and the artificially generated contaminated background data $\{\boldsymbol{D}_j(c_o) + \boldsymbol{E}_j^{\text{N}}\}_{j=0}^{2n-1}$, respectively. Let $\{\sigma_j^o\}_{j=1}^{nm}$ be the singular values of $\boldsymbol{M}(c_o)$, and $\{\sigma_j^{\text{N}}\}_{j=1}^{nm}$ the singular values of $\boldsymbol{M}^{\text{N}}(c_o)$, sorted in decreasing order. Choose a small ε_σ, the largest relative deviation of singular values past which we consider them contaminated by noise. Let R^{N} be the smallest among j such that

$$\left| \frac{\sigma_j^{\text{N}}}{\sigma_j^o} - 1 \right| \ge \varepsilon_\sigma. \tag{D-2}$$

Then, we can estimate $r = \lfloor R^{\text{N}}/m \rfloor$.

Note that the estimation can be adaptive: we can choose at iteration i in Algorithm 2 the value r_i obtained as above but with $\boldsymbol{M}(v(\cdot;\boldsymbol{\eta}^{(i)}))$ instead of $\boldsymbol{M}(c_o)$. However, in our examples this was not necessary, since using $\boldsymbol{M}(c_o)$ provided a robust if somewhat conservative estimate, as shown in the numerical example described below.

In Fig. D.1 we illustrate the choice of regularization threshold for the Marmousi model in the setting outlined in the numerical results section ($m = 30$, $n = 40$, 1% additive noise). The left plot shows the singular values σ_j^o and σ_j^{N} for a range $j = 900, 901, \ldots, 1025$, while also comparing them to the singular values of $\boldsymbol{M}$ and $\boldsymbol{M}^{\text{N}}$. Setting $\varepsilon_\sigma = 10^{-2}$, we obtain $R^{\text{N}} = 944$ from (D-2), as shown in the right plot. This gives the value $r = \lfloor 944/30 \rfloor = 31$

used in the numerical experiments. Note that this process estimates well the point after which the singular values of $\boldsymbol{M}^{\mathrm{N}}$ diverge from those of $\boldsymbol{M}$, as seen in the left plot.

REFERENCES

Borcea, L., V. Druskin, A. Mamonov, and M. Zaslavsky, 2019, Robust nonlinear processing of active array data in inverse scattering via truncated reduced order models: Journal of Computational Physics, **381**, 1–26.

Borcea, L., V. Druskin, A. Mamonov, M. Zaslavsky, and J. Zimmerling, 2020, Reduced order model approach to inverse scattering: SIAM Journal on Imaging Sciences, **13**, 685–723.

Borcea, L., V. Druskin, A. V. Mamonov, and M. Zaslavsky, 2018, Untangling the nonlinearity in inverse scattering with data-driven reduced order models: Inverse Problems, **34**, 065008.

Borcea, L., J. Garnier, A. Mamonov, and J. Zimmerling, 2021, Reduced order model approach for imaging with waves: Inverse Problems, **38**, 025004 (40pp).

Bozdağ, E., J. Trampert, and J. Tromp, 2011, Misfit functions for full waveform inversion based on instantaneous phase and envelope measurements: Geophysical Journal International, **185**, 845–870.

Brossier, R., S. Operto, and J. Virieux, 2010, Which data residual norm for robust elastic frequency-domain full waveform inversion?: Geophysics, **75**, R37–R46.

Brunton, S., and J. Kutz, 2019, Data-driven science and engineering: Machine learning, dynamical systems, and control: Cambridge University Press.

Brunton, S., J. Proctor, and J. Kutz, 2016, Discovering governing equations from data by sparse identification of nonlinear dynamical systems: Proceedings of the national academy of sciences, **113**, 3932–3937.

Bunks, C., F. Saleck, S. Zaleski, and G. Chavent, 1995, Multiscale seismic waveform inversion: Geophysics, **60**, 1457–1473.

Clayton, R., and R. Stolt, 1981, A Born-WKBJ inversion method for acoustic reflection data: Geophysics, **46**, 1559–1567.

Dines, K., and R. Lytle, 1979, Computerized geophysical tomography: Proceedings of the IEEE, **67**, 1065–1073.

Druskin, V., A. Mamonov, A. Thaler, and M. Zaslavsky, 2016, Direct, nonlinear inversion algorithm for hyperbolic problems via projection-based model reduction: SIAM Journal on Imaging Sciences, **9**, 684–747.

Druskin, V., A. Mamonov, and M. Zaslavsky, 2018, A nonlinear method for imaging with acoustic waves via reduced order model backprojection: SIAM Journal on Imaging Sciences, **11**, 164–196.

Engquist, B., and B. Froese, 2014, Optimal transport for seismic full waveform inversion: Communications in Mathematical Sciences, **12**, 979–988.

Gauthier, O., J. Virieux, and A. Tarantola, 1986, Two-dimensional nonlinear inversion of seismic waveforms: Numerical results: Geophysics, **51**, 1387–1403.

Golub, G., and C. Van Loan, 2013, Matrix Computations, 4 ed.: The Johns Hopkins University Press.

Herkt, S., M. Hinze, and R. Pinnau, 2013, Convergence analysis of galerkin pod for linear second order evolution equations: Electron. Trans. Numer. Anal, **40**, 321–337.

Hesthaven, J., G. Rozza, and B. Stamm, 2016, Certified reduced basis methods for parametrized partial differential equations: Springer.

Huang, G., R. Nammour, and W. Symes, 2017, Full-waveform inversion via source-receiver extension: Geophysics, **82**, R153–R171.

Kunisch, K., and S. Volkwein, 2010, Optimal snapshot location for computing pod basis functions: ESAIM: Mathematical Modelling and Numerical Analysis, **44**, 509–529.

McLean, W., 2000, Strongly elliptic systems and boundary integral equations: Cambridge university press.

Peherstorfer, B., and K. Willcox, 2016, Data-driven operator inference for nonintrusive projection-based model reduction: Computer Methods in Applied Mechanics and Engineering, **306**, 196–215.

Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain, Part 1: Theory and verification in a physical scale model: Geophysics, **64**, 888–901.

Santosa, F., and W. W. Symes, 1989, Analysis of least-squares velocity inversion: Society of exploration Geophysicists. Geophysical Monograph 4.

Sava, P., and B. Biondi, 2004, Wave-equation migration velocity analysis. i. theory: Geophysical Prospecting, **52**, 593–606.

Stefanov, P., G. Uhlmann, A. Vasy, and H. Zhou, 2019, Travel time tomography: Acta Mathematica Sinica, English Series, **35**, 1085–1114.

Symes, W., 2008, Migration velocity analysis and waveform inversion: Geophysical prospecting, **56**, 765–790.

Symes, W., and J. J. Carazzone, 1991, Velocity inversion by differential semblance optimization: Geophysics, **56**, 654–663.

Symes, W., and M. Kern, 1994, Inversion of reflection seismograms by differential semblance analysis: Algorithm structure and synthetic examples 1: Geophysical Prospecting, **42**, 565–614.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1–WCC26.

Yang, Y., B. Engquist, J. Sun, and B. Hamfeldt, 2018, Application of optimal transport and the quadratic wasserstein metric to full-waveform inversion: Geophysics, **83**, R43–R62.