GEOPHYSICS, VOL. 88, NO. 2 (MARCH-APRIL 2023); P. R175–R191, 10 FIGS., 2 TABLES.
10.1190/GEO2022-0070.1

Waveform inversion via reduced order modeling

Liliana Borcea[1], Josselin Garnier[2], Alexander V. Mamonov[3], and Jörn Zimmerling[4]

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 gathered by an acquisition system that probes the unknown medium with pulses and measures the generated waves. We propose to solve the inverse problem of velocity estimation by minimizing the square misfit between the ROM computed from the recorded data and the ROM computed from the modeled data, at the current guess of the velocity. We give a step by step computation of the ROM, which depends nonlinearly on the data and yet can be obtained from them in a noniterative fashion, using efficient methods from linear algebra. We also explain how to make the ROM robust to data inaccuracy. The ROM computation requires the full array response matrix gathered with colocated sources and receivers. However, we find that the computation can deal with an approximation of this matrix, obtained from towed-streamer data using interpolation and reciprocity on-the-fly. Although the full-waveform inversion approach of nonlinear least-squares data fitting is challenging without low-frequency information, due to multiple minima of the data fit objective function, we find that the ROM misfit objective function has better behavior, even for a poor initial guess. We also find by explicit computation of the objective functions in a simple setting that the ROM misfit objective function has convexity properties, whereas the least-squares data fit objective function displays multiple local minima.

INTRODUCTION

We study the inverse problem of velocity estimation from reflection data gathered by an array of N_s colocated sources and receivers. The methodology applies to any linear wave equation, for 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 $c(\boldsymbol{x})$.

Let $p^{(s)}(t,\boldsymbol{x})$ model the pressure wave generated by the sth source, for $s = 1, \ldots, N_s$. It satisfies the wave equation:

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

$$p^{(s)}(t,\boldsymbol{x}) = 0, \quad t < -t_{\mathrm{f}}, \tag{2}$$

for $\boldsymbol{x} \in \Omega$, a connected domain, with boundary $\partial\Omega$. This domain can arise from the mathematical truncation of the space because 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 impose any homogeneous boundary conditions at $\partial\Omega$, for example, Dirichlet.

The right side in equation 1 models the point-like source at location $\boldsymbol{x}_s$, where $\delta_{\boldsymbol{x}_s}(\boldsymbol{x})$ denotes the Dirac $\delta(\boldsymbol{x} - \boldsymbol{x}_s)$, $f(t)$ is the probing pulse, and the prime stands for the time derivative. It is convenient for the analysis to assume that $f(t)$ is an even function, with support in the interval $(-t_{\mathrm{f}}, t_{\mathrm{f}})$. This may not be the case, in practice, but we explain later that data gathered with an arbitrary pulse that is known or can be estimated, can be transformed by simple processing to data for an even pulse $f(t)$. Prior to the excitation, the medium is quiescent, as stated in equation 2.

Manuscript received by the Editor 3 February 2022; revised manuscript received 3 October 2022; published ahead of production 24 November 2022; published online 22 February 2023.
[1]University of Michigan, Department of Mathematics, Ann Arbor, Michigan, USA. E-mail: borcea@umich.edu.
[2]Institut Polytechnique de Paris, Centre de Mathématiques Appliquées, Ecole Polytechnique, Palaiseau, France. E-mail: josselin.garnier@polytechnique.edu.
[3]University of Houston, Department of Mathematics, Houston, Texas, USA. E-mail: mamonov@math.uh.edu.
[4]University of Michigan, Department of Mathematics, Ann Arbor, Michigan, USA and Uppsala Universitet, Department of Information Technology, Division of Scientific Computing, Uppsala, Sweden. E-mail: jtzimmerling@gmail.com (corresponding author).

DOI:10.1190/geo2022-0070.1

The inverse problem is to find the velocity $c(\boldsymbol{x})$ from the measured array response matrix $\boldsymbol{\mathcal{M}}(t)$, with entries

$$\mathcal{M}^{(r,s)}(t) = p^{(s)}(t, \boldsymbol{x}_r), \quad 1 \le r, s \le N_s, t \in (-t_f, T]. \quad (3)$$

Note that knowing $\boldsymbol{\mathcal{M}}(t)$ requires colocated sources and receivers. This is typically not the case in geophysics applications, but the formulation extends, for example, to the towed-streamer data acquisition. The missing off-diagonal entries of $\boldsymbol{\mathcal{M}}(t)$ are obtained from towed-streamer data using source-receiver reciprocity on-the-fly, and the diagonal entries can be approximated by interpolation.

Common velocity estimation approaches are traveltime tomography (Dines and Lytle, 1979) and its more general version studied in the mathematics community (Stefanov et al., 2019), linearized, also known as the Born inversion (Clayton and Stolt, 1981), migration velocity analysis (Symes and Carazzone, 1991; Sava and Biondi, 2004), and full-waveform inversion (FWI) (Tarantola, 1984; Virieux and Operto, 2009). The first three are based on assumptions such as the velocity changes slowly on the scale of the wavelength (for traveltime tomography) or the velocity variations are small (for the Born inversion) or there is a separation of scales between the smooth components of the velocity and the rough part that gives the reflectivity of the medium (for migration). FWI circumvents such assumptions. It is a partial differential equation constrained optimization that fits the data with its model prediction, typically in the L^2 (least-squares) sense. The increase in computing power has led to growing interest in FWI, but there is a fundamental impediment, which manifests especially for high-frequency data. The objective function is nonconvex even in the absence of noise (Gauthier et al., 1986; Santosa and Symes, 1989) and displays numerous local minima. This issue, which is due to nonlinear (multiple scattering) effects and cycle skipping, makes any gradient-based, 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. For instance, 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), such as 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 also are approaches that use a better alternative than the L^2 norm for measuring the data misfit (Guitton and Symes, 2003; Brossier et al., 2010; Bozdağ et al., 2011). A prominent alternative is an optimal transport (Wasserstein) metric proposed and analyzed for seismic inversion in Engquist and Froese (2014) and 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 defined in equation 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 for a poor initial guess.

There is an ever-growing 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. 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 instances and for all $\boldsymbol{x} \in \Omega$. In contrast, seismic surveys only provide the measurements $\boldsymbol{\mathcal{M}}(t)$ of the wave at the receiver positions.

The first array of 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 wavefield from one instance to the next one and 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, 2021). 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 has convexity properties. This is not the case for the FWI misfit objective function, 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 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 a general description that motivates our ROM-based approach to velocity estimation and gives the key ideas behind the ROM construction. Then, we discuss the mathematical details that establish the relationship between the ROM and the data and we summarize the ROM construction in the form of an algorithm. The methodology introduced in this section assumes noiseless data and full knowledge of the array response matrix $\boldsymbol{\mathcal{M}}(t)$. This allows us to describe the objective function for velocity estimation without using a penalty regularization term. However, regularization is important and must be done carefully, as explained later in the paper, when dealing with noisy data and the approximation of $\boldsymbol{\mathcal{M}}(t)$ from the towed-streamer type of measurements.

Outline and motivation of the method

The FWI approach to velocity estimation seeks an approximate inverse of the nonlinear forward map $c(\boldsymbol{x}) \overset{\mathcal{F}}{\mapsto} \boldsymbol{\mathcal{M}}(t)$ using the data misfit minimization:

$$\min_{v \in \mathcal{C}} \int_{-t_f}^{T} dt \, \|\boldsymbol{\mathcal{M}}(t) - \mathcal{F}[v](t)\|_F^2 + \text{regularization}, \quad (4)$$

where v denotes the search velocity in the search space $\mathcal{C}$ and $\|\cdot\|_F$ is the matrix Frobenius norm. Our approach introduces an additional mapping from $\boldsymbol{\mathcal{M}}(t)$ to an approximation of the symmetrized wave operator $\partial_t^2 + \mathcal{A}$. The symmetrization is carried out with a similarity transformation of the usual wave operator $\partial_t^2 - c^2(\boldsymbol{x})\Delta$. It amounts to scaling $p^{(s)}(t, \boldsymbol{x})$ by $c^{-1}(\boldsymbol{x})$ and gives

$$\mathcal{A} = c^{-1}(\boldsymbol{x})[-c^2(\boldsymbol{x})\Delta]c(\boldsymbol{x}) = -c(\boldsymbol{x})\Delta[c(\boldsymbol{x})\,\cdot]. \quad (5)$$

The approximation that we seek is the ROM wave operator $\partial_t^2 + \boldsymbol{\mathcal{A}}^{\text{ROM}}$, where $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ is a symmetric and positive definite matrix, a Galerkin approximation of the self-adjoint and positive

definite operator $\mathcal{A}$. Roughly speaking, we can think of the data to ROM mapping $\mathcal{R}$ as a preconditioner of the forward mapping $\mathcal{F}$

$$c(\boldsymbol{x}) \overset{\mathcal{F}}{\mapsto} \boldsymbol{\mathcal{M}}(t) \overset{\mathcal{R}}{\mapsto} \boldsymbol{\mathcal{A}}^{\text{ROM}}, \tag{6}$$

because the composition $\mathcal{R} \circ \mathcal{F}$, which gives $\boldsymbol{\mathcal{A}}^{\text{ROM}} = \mathcal{R}(\mathcal{F}[c])$, is easier to "invert."

The Galerkin method is a standard way of approximating an operator, like $\mathcal{A}$, by a matrix. Typically, the approximation is in spaces of piecewise polynomial functions with support over a few grid cells (Brenner and Scott, 2008). If we gather these functions in a row vector field $\boldsymbol{\Psi}(\boldsymbol{x})$, the matrix approximation of $\mathcal{A}$ is

$$\boldsymbol{\mathcal{A}}^{\Psi} = \int_{\Omega} d\boldsymbol{x}\, \boldsymbol{\Psi}^{T}(\boldsymbol{x}) \mathcal{A} \boldsymbol{\Psi}(\boldsymbol{x}). \tag{7}$$

This matrix $\boldsymbol{\mathcal{A}}^{\Psi}$ has a much simpler dependence on $c(\boldsymbol{x})$ than $\boldsymbol{\mathcal{M}}(t) = \mathcal{F}[c](t)$ because its entries depend quadratically on the coefficient $c(\boldsymbol{x})$ integrated locally, in a few grid cells. It would be easy to find $c(\boldsymbol{x})$ from $\boldsymbol{\mathcal{A}}^{\Psi}$, but this matrix cannot be computed from the measurements $\boldsymbol{\mathcal{M}}(t)$.

Our ROM matrix $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ is a Galerkin approximation of $\mathcal{A}$ on the space spanned by the snapshots of the wavefield, at N_{t} discrete and equidistant time instances. Such approximation spaces are common in model order reduction (Hesthaven et al., 2016; Brunton and Kutz, 2019), where the idea is to use the history of the wavefield to extrapolate or interpolate its behavior. Our projection of $\mathcal{A}$ is carried out using an orthonormal basis of the space of snapshots, gathered in the row vector field $\mathbf{V}(\boldsymbol{x})$,

$$\boldsymbol{\mathcal{A}}^{\text{ROM}} = \int_{\Omega} d\boldsymbol{x}\, \boldsymbol{V}^{T}(\boldsymbol{x}) \mathcal{A} \boldsymbol{V}(\boldsymbol{x}) \in \mathbb{R}^{N_{\text{t}}N_{\text{s}} \times N_{\text{t}}N_{\text{s}}}. \tag{8}$$

Here, the important observations about $\boldsymbol{V}(\boldsymbol{x})$ are as follows:

1) The ROM matrix $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ can be obtained directly from the measurements $\boldsymbol{\mathcal{M}}(t)$, without knowing the snapshots $\boldsymbol{V}(\boldsymbol{x})$ or the operator $\mathcal{A}$. This is one of the most striking results of this paper. We summarize the transform $\mathcal{R}$ from $\boldsymbol{\mathcal{M}}(t)$ to $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ in Algorithm 1 and we explain the relationship between the ROM and the data that leads to Algorithm 1 in the next subsection.
2) The term $\boldsymbol{V}(\boldsymbol{x})$ cannot be computed from the measurements. However, the analysis in Borcea et al. (2021, Appendix A) and the numerical studies in Borcea et al. (2021, Section 6.3) suggest that $\boldsymbol{V}(\boldsymbol{x})$ is almost independent of the rough part of $c(\boldsymbol{x})$, i.e., the reflectivity.
3) The basis functions in $\boldsymbol{V}(\boldsymbol{x})$ associated with the jth time instance are peaked near the maximum depth reached by the wavefield up to this instance.
4) The term $\boldsymbol{V}(\boldsymbol{x})$ is causal. With the first $k < N_{\text{t}}$ snapshots, the definition in equation 8 gives the principal $kN_{\text{s}} \times kN_{\text{s}}$ submatrix of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$, obtained by removing its last $(N_{\text{t}} - k)N_{\text{s}}$ rows and columns.

Because $\boldsymbol{V}(\boldsymbol{x})$ depends on $c(\boldsymbol{x})$ in a complicated way, we cannot prove the convexity of the ROM misfit objective function $v \mapsto \|\boldsymbol{\mathcal{A}}^{\text{ROM}} - \mathcal{R}(\mathcal{F}[v])\|_F^2$ for a general medium. It is only in layered media that the result follows from the proof in Borcea et al. (2021, Appendix A). Explicitly, it is proved there that in a layered medium with variable wave speed and density, containing multiple reflectors of arbitrary strength, the orthonormal basis written in traveltime coordinates is almost the same as the one in a homogeneous medium. This means that at least in the vicinity of the right kinematics, the dependence of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ on $c(\boldsymbol{x})$ is mainly through $\mathcal{A}$, and the objective function is locally convex.

In general media, we expect that, for a rich enough space of snapshots, which allows a good approximation of $\Psi(\boldsymbol{x})$ in equation 7 in terms of $\boldsymbol{V}(\boldsymbol{x})$, the ROM matrix $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ contains roughly the same information as $\boldsymbol{\mathcal{A}}^{\Psi}$. The numerical study in Borcea et al. (2021, Section 6.3) shows that "rich enough" means for sources/receivers separated by roughly half a wavelength and for time sampling satisfying the Nyquist criterium. The third attribute of $\boldsymbol{V}(\boldsymbol{x})$ listed previously and equation 8 also show that the entries of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ depend mostly on the locally integrated $c(\boldsymbol{x})$, similar to $\boldsymbol{\mathcal{A}}^{\Psi}$. Thus, we expect that the velocity estimation from the computable $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ behaves similarly to that from the uncomputable $\boldsymbol{\mathcal{A}}^{\Psi}$, which is why we propose using the minimization:

$$\min_{v \in \mathcal{C}} \|\boldsymbol{\mathcal{A}}^{\text{ROM}} - \mathcal{R}(\mathcal{F}[v])\|_F^2 + \text{regularization}. \tag{9}$$

The minimization problem 9 can be solved with a Gauss-Newton iterative method that is summarized in Algorithm 2. However, first, we explain the relationship between the ROM and the data.

Relationship between the ROM and the data

We begin by transforming equation 1 to a homogeneous wave equation for a new wave $u^{(s)}(t, \boldsymbol{x})$, with an initial state determined by the source. This new wave is defined in the next section and the transformation involves working with the even in time wave

$$p_e^{(s)}(t, \boldsymbol{x}) = [p^{(s)}(t, \boldsymbol{x}) + p^{(s)}(-t, \boldsymbol{x})], \tag{10}$$

where $p^{(s)}(t, \boldsymbol{x})$ solves equations 1 and 2. We can think of the transformation as a Duhamel principle, although it is not in the usual form (John, 1982) because at $t = 0$ we obtain

$$u^{(s)}(0, \boldsymbol{x}) = u_0^{(s)}(\boldsymbol{x}), \quad \partial_t u^{(s)}(0, \boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega, \tag{11}$$

with $u_0^{(s)}(\boldsymbol{x})$ determined by the source location $\boldsymbol{x}_s$ and the probing pulse $f(t)$.

Note that we do not lose any information by working with the even wave in equation 10 and therefore the simple initial conditions in equation 11, as long as we know the medium near the co-located sources/receivers. Near means within the distance of travel over the small time interval $(-t_{\text{f}}, t_{\text{f}})$ of support of $f(t)$. We assume henceforth that the medium near the colocated sources/receivers is known and homogeneous, with velocity $\bar{c}$. Due to the initial condition in equation 2, we observe that

$$p_e^{(s)}(t, \boldsymbol{x}_r) = p^{(s)}(t, \boldsymbol{x}_r), \qquad \text{for } t \geq t_{\text{f}}, \quad s, r = 1, \ldots, N_{\text{s}}. \tag{12}$$

The waves differ at $t \in [0, t_{\text{f}})$, but because for such time the measurements are insensitive to the unknown part of the medium, no information is lost.

Note also that the measurements $p_e^{(s)}(t, \boldsymbol{x}_r)$ are obtained easily from those of $p^{(s)}(t, \boldsymbol{x}_r)$, if the latter are gathered at $t \geq -t_{\text{f}}$, for $s, r = 1, \ldots, N_{\text{s}}$, as assumed in equation 3. But even if the measure-

ments are made at $t \geq t_f$ only, we can still compute $p_e^{(s)}(t, \boldsymbol{x}_r)$ at $t \in [0, t_f)$ by solving the wave equation with velocity $\bar{c}$. Thus, in either case, we can map the measured $\boldsymbol{\mathcal{M}}(t)$ to a new data matrix $\boldsymbol{D}(t)$, with entries at $t \geq 0$ given by

$$D^{(r,s)}(t) = p^{(s)}(t, \boldsymbol{x}_r) + p^{(s)}(-t, \boldsymbol{x}_r) = \boldsymbol{\mathcal{M}}^{(r,s)}(t) + \boldsymbol{\mathcal{M}}^{(r,s)}(-t), \quad 1 \leq r, s \leq N_s. \tag{13}$$

To define our Galerkin approximation space, let us consider a time discretization $t_j = j\tau$, with uniform stepping τ, for $j \geq 0$. We gather the waves $u^{(s)}(t, \boldsymbol{x})$ evaluated at t_j, for all the N_s sources, in the jth snapshot vector field:

$$\boldsymbol{u}_j(\boldsymbol{x}) = (u^{(1)}(t_j, \boldsymbol{x}), \ldots, u^{(N_s)}(t_j, \boldsymbol{x})), \quad \boldsymbol{x} \in \Omega. \tag{14}$$

We are interested only in the first N_t snapshots and organize them in the $N_t N_s$-dimensional row vector field:

$$\boldsymbol{U}(\boldsymbol{x}) = (\boldsymbol{u}_0(\boldsymbol{x}), \ldots, \boldsymbol{u}_{N_t-1}(\boldsymbol{x})), \quad \boldsymbol{x} \in \Omega. \tag{15}$$

The space spanned by the components of $\boldsymbol{U}(\boldsymbol{x})$, denoted range$(\boldsymbol{U}(\boldsymbol{x}))$, is our approximation space and the Galerkin approximation of the wavefield is

$$\boldsymbol{u}_G(t, \boldsymbol{x}) = (u_G^{(1)}(t, \boldsymbol{x}), \ldots, u_G^{(N_s)}(t, \boldsymbol{x})) = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{g}(t), \tag{16}$$

with time-dependent coefficients gathered in the matrices $\boldsymbol{g}(t) \in \mathbb{R}^{N_t N_s \times N_s}$. These coefficients are such that when substituting equation 16 into the homogeneous wave equation, the residual is orthogonal to the approximation space. This gives the following system of second-order ordinary differential equations:

$$\underbrace{\int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x})}_{\boldsymbol{M}} \boldsymbol{g}''(t) + \underbrace{\int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\mathcal{A}\boldsymbol{U}(\boldsymbol{x})}_{\boldsymbol{S}} \boldsymbol{g}(t) = 0, \tag{17}$$

for $t > 0$, with the initial condition:

$$\boldsymbol{g}(0) = \boldsymbol{e}_0, \quad \boldsymbol{g}'(0) = \boldsymbol{0}, \tag{18}$$

where $\boldsymbol{e}_0$ is the first $N_t N_s \times N_s$ block of the $N_t N_s \times N_t N_s$ identity matrix $\boldsymbol{I}_{N_t N_s}$. Equation 18 ensures that the Galerkin approximation 16 satisfies the initial conditions:

$$\boldsymbol{u}_G(0, \boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{e}_0 = \boldsymbol{u}_0(\boldsymbol{x}), \quad \partial_t \boldsymbol{u}_G(0, \boldsymbol{x}) = \boldsymbol{0}. \tag{19}$$

The Galerkin approximation described previously would be straightforward if we knew $\boldsymbol{U}(\boldsymbol{x})$, but we do not know it. Our key observation is that the $N_t N_s \times N_t N_s$ Gramian matrix

$$\boldsymbol{M} = \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x}) \in \mathbb{R}^{N_t N_s \times N_t N_s}, \tag{20}$$

called the "mass matrix" in Galerkin jargon, and the "stiffness matrix"

$$\boldsymbol{S} = \int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\mathcal{A}\boldsymbol{U}(\boldsymbol{x}) \in \mathbb{R}^{N_t N_s \times N_t N_s}, \tag{21}$$

can be calculated directly from $\boldsymbol{D}(t)$ and the second derivative $\boldsymbol{D}''(t)$, evaluated at instances $\{t_j = j\tau\}_{j=0}^{2N_t-2}$, as explained in the next section (for the estimation of $\boldsymbol{D}''(t)$, using a filtered Fourier transform, see Appendix A). Thus, even though we do not know the operator $\mathcal{A}$ and the vector field $\boldsymbol{U}(\boldsymbol{x})$, we can compute the Galerkin coefficients $\boldsymbol{g}(t)$ for all $t \geq 0$, by solving the system of equations 17 with the data-driven $\boldsymbol{M}$ and $\boldsymbol{S}$, and the initial conditions given in equation 18.

The final step of the ROM construction is to put equation 17 in an algebraic form that describes the evolution of a causal wave $\boldsymbol{u}^{\mathrm{ROM}}(t) \in \mathbb{R}^{N_t N_s \times N_s}$. Each column of this wave corresponds to a source index s, with $1 \leq s \leq N_s$. Initially, the true wave is supported near the sources, which is reflected in the algebraic structure of $\boldsymbol{u}^{\mathrm{ROM}}(0)$, whose only nonzero entries are in the first $N_s \times N_s$ block. At later times, there is block row fill-in in $\boldsymbol{u}^{\mathrm{ROM}}(t)$, which models wave propagation further away from the sources.

The desired transformation of equation 17 is achieved using the block Cholesky square root (Golub and Van Loan, 2013) of the data-driven mass matrix

$$\boldsymbol{M} = \boldsymbol{R}^T\boldsymbol{R}, \tag{22}$$

where $\boldsymbol{R}$ is the block upper triangular (with blocks of size $N_s \times N_s$). The wave in the ROM space is defined by

$$\boldsymbol{u}^{\mathrm{ROM}}(t) = \boldsymbol{R}\boldsymbol{g}(t), \tag{23}$$

and we note from equation 18 that at $t = 0$ it satisfies

$$\boldsymbol{u}^{\mathrm{ROM}}(0) = \boldsymbol{R}\boldsymbol{e}_0 = \begin{pmatrix} \boldsymbol{R}_{0,0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix}, \quad \frac{d\boldsymbol{u}^{\mathrm{ROM}}}{dt}(0) = \boldsymbol{0}, \tag{24}$$

where $\boldsymbol{R}_{0,0} \in \mathbb{R}^{N_s \times N_s}$. The wave equation in the ROM space is obtained after multiplying equation 17 on the left by $\boldsymbol{R}^{-T} = (\boldsymbol{R}^{-1})^T$,

$$\frac{d^2\boldsymbol{u}^{\mathrm{ROM}}}{dt^2}(t) + \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}\boldsymbol{u}^{\mathrm{ROM}}(t) = 0, \quad t > 0, \tag{25}$$

and the ROM approximation of $\mathcal{A}$ is the $N_t N_s \times N_t N_s$ matrix:

$$\boldsymbol{\mathcal{A}}^{\mathrm{ROM}} = \boldsymbol{R}^{-T}\boldsymbol{S}\boldsymbol{R}^{-1}. \tag{26}$$

Note that the same block upper triangular matrix $\boldsymbol{R}$ arises in the Gram-Schmidt orthogonalization of the components of $\boldsymbol{U}(\boldsymbol{x})$ given by

$$\boldsymbol{U}(\boldsymbol{x}) = \boldsymbol{V}(\boldsymbol{x})\boldsymbol{R}, \tag{27}$$

where $\boldsymbol{V}(\boldsymbol{x})$ is an $N_t N_s$-dimensional row vector field, with orthonormal components, i.e., it satisfies

$$\int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{I}_{N_t N_s}. \tag{28}$$

The $\boldsymbol{V}(\boldsymbol{x})$ stores the orthonormal basis mentioned earlier in the section. Its causality, in the sense that the jth (m-dimensional) component of $\boldsymbol{V}(\boldsymbol{x})$ is determined by $\boldsymbol{u}_0(\boldsymbol{x}), \ldots, \boldsymbol{u}_j(\boldsymbol{x})$, is built into the

Gram-Schmidt orthogonalization procedure, and therefore in the block upper triangular structure of $\boldsymbol{R}$. Substituting equation 27 into equation 20, and using equation 28, we observe that $\boldsymbol{R}$ in equation 27 is the same as in equation 22 because

$$\boldsymbol{M} = \boldsymbol{R}^T \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x})\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{R}. \tag{29}$$

If we use the Gram-Schmidt equation 27 in equation 26 and recall equation 21 for $\boldsymbol{S}$, we obtain that $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ satisfies equation 8. Therefore, the data-driven $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ defined in equation 26, is in fact the orthogonal projection of the operator $\mathcal{A}$ on the unknown space range$(\boldsymbol{U}(\boldsymbol{x}))$, obtained with the unknown causal and orthonormal basis in $\boldsymbol{V}(\boldsymbol{x})$.

We can now add a fifth observation about $\boldsymbol{V}(\boldsymbol{x})$. It has been proved recently in Borcea et al. (2022, Proposition 3.2) that the snapshots gathered in $\boldsymbol{U}(\boldsymbol{x};v) = \boldsymbol{V}(\boldsymbol{x};v)\boldsymbol{R}$ satisfy exactly the data $\{\boldsymbol{D}(j\tau)\}_{j=0}^{2n-2}$. The difference between this field and the true one in equation 27 is that the unknown $\boldsymbol{V}(\boldsymbol{x})$ is replaced by $\boldsymbol{V}(\boldsymbol{x};v)$, whose components are the orthonormal basis functions computed with the guess velocity $v(\boldsymbol{x})$. Any guess velocity works, even $v(\boldsymbol{x}) = \bar{c}$. That $\boldsymbol{U}(\boldsymbol{x})$ and $\boldsymbol{U}(\boldsymbol{x};v)$ give an exact data fit, means that the data-driven matrix $\boldsymbol{R}$ contains all the information. This is why, as shown in Borcea et al. (2022), $\boldsymbol{U}(\boldsymbol{x};v)$ contains all the arrival events present in $\boldsymbol{U}(\boldsymbol{x})$. The purpose of $\boldsymbol{V}(\boldsymbol{x})$ in equation 27 may be viewed as mapping the information in $\boldsymbol{R}$, from the algebraic (ROM) space to the physical space. When we have the incorrect kinematics (smooth part of $v(\boldsymbol{x})$), $\boldsymbol{V}(\boldsymbol{x};v)$ maps the arrivals to incorrect depths. However, if the kinematics is only slightly incorrect, the computable $\boldsymbol{V}(\boldsymbol{x};v)$ is very close to the uncomputable $\boldsymbol{V}(\boldsymbol{x})$. This is another way of explaining that at least close enough to the true velocity, $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ defined in equation 8 depends on $c(\boldsymbol{x})$ mostly through $\mathcal{A}$ and the objective function of the ROM misfit is locally convex.

Technical details of ROM computation

The flow chart of the computation of $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ from the measurements $\boldsymbol{\mathcal{M}}(t)$ is shown in Figure 1. The first step computes the data matrices

$$\boldsymbol{D}(t) = (\mathcal{M}^{(r,s)}(t) + \mathcal{M}^{(r,s)}(-t))_{r,s=1}^{N_s}, \tag{30}$$

and their second derivatives $\boldsymbol{D}''(t)$ at instances $t = j\tau$, for $0 \le j \le 2N_t - 2$. Recall from the previous discussion that $\mathcal{M}^{(r,s)}(-t)$ contributes only at $t = j\tau \in [0, t_f)$ and it may either be measured or computed in the reference medium with velocity $\bar{c}$. The details on the computation of the second derivative $\boldsymbol{D}''(t)$ are given in Appendix A. Consistent with our previous notation convention, we denote henceforth

$$\boldsymbol{D}_j = \boldsymbol{D}(j\tau), \quad \ddot{\boldsymbol{D}}_j = \boldsymbol{D}''(j\tau). \tag{31}$$

Before we explain the second step in the flow chart, let us give a few technical details of the definition of the new wave $u^{(s)}(t,\boldsymbol{x})$ and the derivation of the inner product expression in equation 40 of the data matrices. These details are not needed to compute $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$, which is why they are not in the flow chart, but they allow us to derive the expression of the mass and stiffness matrices in terms of the data.

It is proved in Borcea et al. (2020, Appendix A) that

$$\frac{[p^{(s)}(t,\boldsymbol{x}) + p^{(s)}(-t,\boldsymbol{x})]}{c(\boldsymbol{x})/\bar{c}} = \cos(t\sqrt{\mathcal{A}})\hat{f}(\sqrt{\mathcal{A}})\delta_{\boldsymbol{x}_s}(\boldsymbol{x}) = \sum_{j=1}^{\infty} \cos(t\sqrt{\lambda_j})\hat{f}(\sqrt{\lambda_j})y_j(\boldsymbol{x}_s)y_j(\boldsymbol{x}), \tag{32}$$

where

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t)e^{i\omega t}dt \tag{33}$$

is the Fourier transform of the probing pulse and we define functions of the self-adjoint and positive definite operator $\mathcal{A}$ using its spectral decomposition. If $\mathcal{A}$ has the eigenvalues $\{\lambda_j\}_{j\ge1}$ and the eigenfunctions $\{y_j\}_{j\ge1}$, then $\cos(t\sqrt{\mathcal{A}})$ is the operator with eigenvalues $\{\cos(t\sqrt{\lambda_j})\}_{j\ge1}$ and the same eigenfunctions. The operator $\hat{f}(\sqrt{\mathcal{A}})$ is defined similarly. The derivation of equation 32 involves the expansion of the wavefield in the basis $\{y_j(\boldsymbol{x})\}_{j\ge1}$ of eigenfunctions of $\mathcal{A}$ and manipulations of series.

Next, we need the technical assumption that $\hat{f} \ge 0$. This may not be the case in general, but the assumption can be achieved with simple processing as follows. Suppose that the probing pulse is actually some wavelet $\varphi(t)$ that 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 receivers, with

$$f(t) = \varphi(t) \star_t \varphi(-t). \tag{34}$$

Such $f(t)$ is obviously an even function, with Fourier transform $\hat{f}(\omega) = |\hat{\varphi}(\omega)|^2 \ge 0$, that is analytic by the Paley-Wiener-Schwartz theorem (Hörmander, 2003, Chapter VII).

Analytic functions of $\mathcal{A}$ commute, as can be checked using power series, so we can factor the right side in equation 32 as

$$\cos\left(t\sqrt{\mathcal{A}}\right)\hat{f}\left(\sqrt{\mathcal{A}}\right)\delta_{\boldsymbol{x}_s}(\boldsymbol{x}) = \hat{f}^{\frac{1}{2}}\left(\sqrt{\mathcal{A}}\right)u^{(s)}(t,\boldsymbol{x}), \tag{35}$$

where

$$u^{(s)}(t,\boldsymbol{x}) = \cos\left(t\sqrt{\mathcal{A}}\right)u_0^{(s)}(\boldsymbol{x}) \tag{36}$$

is our new wave, with the initial state

$$u_0^{(s)}(\boldsymbol{x}) = \hat{f}^{\frac{1}{2}}\left(\sqrt{\mathcal{A}}\right)\delta_{\boldsymbol{x}_s}(\boldsymbol{x}). \tag{37}$$

Note that $u^{(s)}(t,\boldsymbol{x})$ is just like the wave written in equation 32. The only difference is that it corresponds to a different pulse, with Fourier transform $\hat{f}^{1/2}$ instead of $\hat{f}$.

There are two important consequences of working with $u^{(s)}(t,\boldsymbol{x})$. The first is that by the definition of $\cos(t\sqrt{\mathcal{A}})$, we can use the trigonometric identity

$$\cos((t+\Delta t)\alpha) = 2\cos(\Delta t\alpha)\cos(t\alpha) - \cos((t-\Delta t)\alpha), \tag{38}$$

for $\alpha = \sqrt{\lambda_j}$, with $j \ge 1$, to evolve the wave defined in equation 36 over any interval Δt,

DOI:10.1190/geo2022-0070.1

$$u^{(s)}(t+\Delta t,\boldsymbol{x}) = 2\cos(\Delta t\sqrt{\mathcal{A}})u^{(s)}(t,\boldsymbol{x}) - u^{(s)}(t-\Delta t,\boldsymbol{x}). \tag{39}$$

The second consequence is that the entries of $\boldsymbol{D}(t)$, defined in equation 13, admit a useful symmetric inner product expression:

$$\begin{aligned} D^{(r,s)}(t) &= p^{(s)}(t,\boldsymbol{x}_r) + p^{(s)}(-t,\boldsymbol{x}_r) \\ &= \int_\Omega d\boldsymbol{x}\, \delta_{\boldsymbol{x}_r}(\boldsymbol{x})\hat{f}^{\frac{1}{2}}\left(\sqrt{\mathcal{A}}\right)u^{(s)}(t,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\left[\hat{f}^{\frac{1}{2}}\left(\sqrt{\mathcal{A}}\right)\delta_{\boldsymbol{x}_r}(\boldsymbol{x})\right]u^{(s)}(t,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})u^{(s)}(t,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})\cos\left(t\sqrt{\mathcal{A}}\right)u_0^{(s)}(\boldsymbol{x}), \end{aligned} \tag{40}$$

for $1 \le r, s \le N_s$. The second equality in this equation is from equation 32 and the assumption $c(\boldsymbol{x}_r) = \bar{c}$, the third equality is because $\mathcal{A}$ and therefore $\hat{f}^{1/2}(\sqrt{\mathcal{A}})$ are self-adjoint operators that commute, and the last equalities follow from equations 36 and 37. We also have

$$\begin{aligned} \frac{d^2D^{(r,s)}(t)}{dt^2} &= \int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})\partial_t^2 u^{(s)}(t,\boldsymbol{x}) \\ &= -\int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})\mathcal{A}u^{(s)}(t,\boldsymbol{x}), \quad 1 \le r,s \le N_s. \end{aligned} \tag{41}$$

Now we can describe how we use equations 39–41 to complete the second step in the flow chart of Figure 1. With the notation

$$\langle \boldsymbol{\phi}, \boldsymbol{\psi}\rangle = \int_\Omega d\boldsymbol{x}\, \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}), \tag{42}$$

for the integral of the outer product of any two functions $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\psi}(\boldsymbol{x})$ with values in $\mathbb{R}^{1\times N_s}$, and from the definition in equation 20, we compute the $N_s \times N_s$ blocks of the mass matrix as

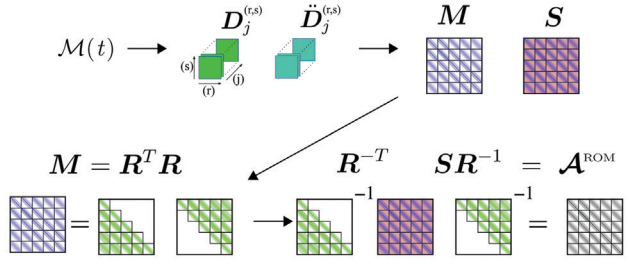

Figure 1. Flow chart for the computation of the ROM from the measurements. There are four steps, each indicated with an arrow. All of the matrices are of size $N_tN_s \times N_tN_s$, with entries organized in $N_s \times N_s$ blocks.

$$\begin{aligned} \boldsymbol{M}_{i,j} &= \langle \boldsymbol{u}_i, \boldsymbol{u}_j\rangle = \left\langle \cos\left(i\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0, \cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0\right\rangle \\ &= \left\langle \boldsymbol{u}_0, \cos\left(i\tau\sqrt{\mathcal{A}}\right)\cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0\right\rangle \\ &= \frac{1}{2}\left\langle \boldsymbol{u}_0, \left[\cos\left((i+j)\tau\sqrt{\mathcal{A}}\right) + \cos\left(|i-j|\tau\sqrt{\mathcal{A}}\right)\right]\boldsymbol{u}_0\right\rangle \\ &= \frac{1}{2}(\boldsymbol{D}_{i+j} + \boldsymbol{D}_{|i-j|}), \quad 0 \le i,j \le N_t - 1. \end{aligned} \tag{43}$$

The second line in this equation is because $\mathcal{A}$ and therefore $\cos(i\tau\sqrt{\mathcal{A}})$ are self-adjoint operators that commute, the third line is due to equation 39, evaluated at $t = i\tau$ and $\Delta t = j\tau$, and the last line is in equation 40. The blocks of the stiffness matrix defined in equation 21 are

$$\begin{aligned} \boldsymbol{S}_{i,j} &= \langle \boldsymbol{u}_i, \mathcal{A}\boldsymbol{u}_j\rangle = \left\langle \cos\left(i\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0, \mathcal{A}\cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0\right\rangle \\ &= \left\langle \boldsymbol{u}_0, \mathcal{A}\cos\left(i\tau\sqrt{\mathcal{A}}\right)\cos\left(j\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_0\right\rangle \\ &= \frac{1}{2}\langle \boldsymbol{u}_0, \mathcal{A}\boldsymbol{u}_{i+j} + \mathcal{A}\boldsymbol{u}_{|i-j|}\rangle \\ &= -\frac{1}{2}(\ddot{\boldsymbol{D}}_{i+j} + \ddot{\boldsymbol{D}}_{|i-j|}), \quad 0 \le i,j \le N_t - 1, \end{aligned} \tag{44}$$

where we used again the self-adjointness of $\mathcal{A}$, and equation 39 evaluated at $t = i\tau$ and $\Delta t = j\tau$. The last equality is in equation 41. The block structure of the matrices $\boldsymbol{M}$ and $\boldsymbol{S}$ is sketched in Figure 1 for the case $N_t = 5$.

The remaining two steps in the flow chart in Figure 1 are self-explanatory and have been motivated in the previous subsection. We summarize the computation of $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ in the following algorithm.

ROM-based velocity estimation

We estimate $c(\boldsymbol{x})$ by minimizing the misfit of the ROM, as shown in equation 9. The computation of the term $\mathcal{R}(\mathcal{F}[v])$ in that equation involves two steps. The first step is to solve the wave equation 1 with $c(\boldsymbol{x})$ replaced by the search velocity $v(\boldsymbol{x})$. The solution evaluated at the receivers gives $\mathcal{F}[v](t)$. The second step is to apply Algorithm 1 with input $\mathcal{F}[v](t)$. In an abuse of notation, we let henceforth

$$\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v) = \mathcal{R}(\mathcal{F}[v]). \tag{45}$$

The search space $\mathcal{C}$, where $v(\boldsymbol{x})$ lies, is parameterized using some appropriate basis functions $\{\phi_l(\boldsymbol{x})\}_{l=1}^N$

$$v(\boldsymbol{x};\boldsymbol{\eta}) = c_o(\boldsymbol{x}) + \sum_{l=1}^N \eta_l\phi_l(\boldsymbol{x}), \tag{46}$$

where $c_o(\boldsymbol{x})$ is the initial guess. The optimization is then N dimensional, for the vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_N)^T$ of coefficients in equation 46.

The causality of the ROM (Appendix B) allows us to carry out the inversion in a layer stripping fashion, from the data at time instances $\{t_j = j\tau\}_{j=0}^{2k-2}$, with $k \le N_t$. To do so, we replace

$\mathcal{A}^{\mathrm{ROM}}(v)$ and $\mathcal{A}^{\mathrm{ROM}}$ in the objective function by the upper left $kN_\mathrm{s} \times kN_\mathrm{s}$ blocks of these matrices, denoted by $[\mathcal{A}^{\mathrm{ROM}}(v)]_k$ and $[\mathcal{A}^{\mathrm{ROM}}]_k$, respectively.

Because $\mathcal{A}^{\mathrm{ROM}}$ and thus $[\mathcal{A}^{\mathrm{ROM}}]_k$ are symmetric matrices, it is enough to consider their block upper triangular part in the optimization. As shown in Appendix C, the entries of $\mathcal{A}^{\mathrm{ROM}}$ decay away from the diagonal. Thus, we can ease the computational burden by including only the first few dN_s diagonals in the objective function, where d is an integer between 1 and k. For this purpose, we denote by

$$\mathrm{Rest}_{d,k} : \mathbb{R}^{kN_\mathrm{s} \times kN_\mathrm{s}} \mapsto \mathbb{R}^{dN_\mathrm{s}(kN_\mathrm{s}-(dN_\mathrm{s}-1)/2)}, \tag{47}$$

the mapping that takes a $kN_\mathrm{s} \times kN_\mathrm{s}$ matrix, keeps only its first dN_s upper diagonals, including the main one, and puts their entries into a column vector, of length

$$\sum_{j=0}^{dN_\mathrm{s}-1} (kN_\mathrm{s} - j) = dN_\mathrm{s}[kN_\mathrm{s} - (dN_\mathrm{s} - 1)/2]. \tag{48}$$

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

$$\mathcal{O}_{d,k}(v) = \|\mathrm{Rest}_{d,k}([\mathcal{A}^{\mathrm{ROM}}(v) - \mathcal{A}^{\mathrm{ROM}}]_k)\|_2^2, \tag{49}$$

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

The details on our implementation of Algorithm 2 and the regularization penalty are provided in Appendix D.

Computational cost

Because our Algorithm 2 for ROM-based velocity estimation uses a Gauss-Newton iteration to minimize the objective function in equation 50, we compare its cost to that of the Gauss-Newton method for minimizing the FWI objective function in equation 4. The same parameterization of the search velocity is assumed for both approaches.

The numerical examples considered next are for 2D media $\Omega \subset \mathbb{R}^2$ with a relatively modest number N_s of colocated sources/receivers, not exceeding 60. In such settings, the cost of each Gauss-Newton step is dominated by the computation of the Jacobian of the objective function. This computation requires solving the forward problem for all N_s sources. The ROM-based approach requires, in addition, the computation of $\mathcal{A}^{\mathrm{ROM}}$ and its derivatives. We compare next the cost of solving the forward problem with that of computing the ROM with Algorithm 1.

We solve the forward problem (equations 1 and 2) in a rectangular domain Ω, with homogeneous Dirichlet boundary conditions at $\partial\Omega$, using explicit time stepping, a three-point finite-difference approximation of ∂_t^2 with step τ_f, and a five-point finite-difference discretization of the Laplacian on a uniform mesh with N_f points. To write down the order of N_f, let $\bar{\lambda}$ be the reference wavelength, calculated with the constant reference speed $\bar{c}$ and at the central frequency of the probing signal $f(t)$. An accurate and stable forward solver requires a mesh size h that is a small fraction of the wavelength and does not exceed $\bar{c}\tau_\mathrm{f}$. The number of mesh points is therefore

$$N_\mathrm{f} = \frac{\mathrm{area}(\Omega)}{h^2} \gg N_\mathrm{s}N_\mathrm{t}, \tag{51}$$

where the inequality is because the colocated sources/receivers are at $O(\bar{\lambda})$ distance, the array length is $O(m\bar{\lambda})$ which is usually much smaller than the width of Ω, and the time sample τ used in the ROM construction is much larger than τ_f. Each time step requires multiplying an $N_\mathrm{f} \times N_\mathrm{f}$ sparse matrix with a vector in $\mathbb{R}^{N_\mathrm{f}}$, at an $O(N_\mathrm{f})$ cost.

Algorithm 1. Data-driven ROM operator.

Input: The matrix $\boldsymbol{\mathcal{M}}(t)$ of measurements given in equation 3, at time instances $t = j\tau$, for $j = -N_\mathrm{f}, \ldots, 2N_\mathrm{t} - 2$, with $N_\mathrm{f} = [t_\mathrm{f}/\tau]$. We have $\boldsymbol{\mathcal{M}}(j\tau) = 0$ for $j < -N_\mathrm{f}$.

1) Compute

$$\boldsymbol{D}_j = \boldsymbol{\mathcal{M}}(j\tau) + \boldsymbol{\mathcal{M}}(-j\tau), \quad 0 \le j \le 2N_\mathrm{t} - 2.$$

2) Compute $\{\ddot{\boldsymbol{D}}_j\}_{j=0}^{2N_\mathrm{t}-2}$ using, e.g., the Fourier transform (see Appendix A).

3) Calculate $\boldsymbol{M}, \boldsymbol{S} \in \mathbb{R}^{N_\mathrm{s}N_\mathrm{t} \times N_\mathrm{s}N_\mathrm{t}}$ with the block entries

$$\boldsymbol{M}_{i,j} = (1/2)(\boldsymbol{D}_{i+j} + \boldsymbol{D}_{|i-j|}) \in \mathbb{R}^{N_\mathrm{s} \times N_\mathrm{s}},$$
$$\boldsymbol{S}_{i,j} = -(1/2)(\ddot{\boldsymbol{D}}_{i+j} + \ddot{\boldsymbol{D}}_{|i-j|}) \in \mathbb{R}^{N_\mathrm{s} \times N_\mathrm{s}},$$

for $0 \le i, j \le N_\mathrm{t} - 1$.

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

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

Algorithm 2. ROM-based velocity estimation.

Input: The data-driven $\mathcal{A}^{\mathrm{ROM}}$.

1) Set the number of layers for the layer stripping approach to ℓ and the number of iterations per layer to n_{iter}.

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

$$1 \le k_1 \le k_2 \le \cdots \le k_\ell = N_\mathrm{t}.$$

The data subset for the lth 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 $j = 1, \ldots, n_{\mathrm{iter}}$, set the update index $i = (l-1)n_{\mathrm{iter}} + j$. Compute $\boldsymbol{\eta}^{(i)}$ as a Gauss-Newton update for minimizing the functional

$$\mathcal{L}_i(\boldsymbol{\eta}) = \mathcal{O}_{d,k_l}(v(\cdot;\boldsymbol{\eta})) + \mathcal{L}_i^{\mathrm{reg}}(\boldsymbol{\eta}), \tag{50}$$

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

Output: The velocity estimate $c^{\mathrm{est}}(\boldsymbol{x}) = v(\boldsymbol{x};\boldsymbol{\eta}^{(\ell n_{\mathrm{iter}})})$.

Thus, the cost of solving the forward problem, for the N_s sources and up to time T, is

$$\text{cost}(\mathcal{F}) = O(N_s n_f N_f), \tag{52}$$

where $n_f = T/\tau_f \gg N_t$. Recall that $\mathcal{F}$ denotes the forward map.

The computational cost of running Algorithm 1 lies mainly in the block Cholesky factorization (see equation 22) and the operator ROM computation from equation 26, where $\boldsymbol{R}^{-1}$ can be calculated by a block-wise backward substitution. Therefore, the cost of computing $\mathcal{A}^{\text{ROM}}$ is estimated at

$$\text{cost}(\mathcal{A}^{\text{ROM}}) = O(N_s^3 N_t^3), \tag{53}$$

and it is typically smaller than $\text{cost}(\mathcal{F})$ if the array is not too large and we sample in time at about the Nyquist rate, as explained next, after equation 54. The bulk of the computational cost of derivatives of $\mathcal{A}^{\text{ROM}}$ is in the differentiation of the block Cholesky factor $\boldsymbol{R}$. This cost is essentially the same as that of the block Cholesky factorization itself because the derivatives of $\boldsymbol{R}$ can be computed by a similar factorization algorithm, as described in detail in Borcea et al. (2014, Appendix A).

For 3D media $\Omega \subset \mathbb{R}^3$ and settings with large N_s, the dominant computational cost is not in the Jacobian calculation itself, but in solving the regularized normal equations for the Gauss-Newton update direction for the objective function in equation 50. Although small-scale examples allow for direct computation of the update direction using, e.g., equation D-5, large-scale settings call for iterative approaches such as the conjugate gradient method. However, note that in such settings the computational cost difference between the conventional FWI and ROM-based velocity estimation virtually disappears because the sizes of the Jacobians of both methods can be made essentially identical by an appropriate choice of parameter d in equation 49.

NUMERICAL ILLUSTRATION

In this section, we give two numerical illustrations of the benefits of velocity estimation with the ROM operator versus FWI. We assume, as in the "Theory" section, knowledge of the noiseless array response matrix $\mathcal{M}(t)$. Noisy measurements and the approximation of $\mathcal{M}(t)$ from towed-streamer data are considered in the next section.

The first illustration is for a two-parameter velocity model, where we can plot the objective function over the search space. The second illustration is for the "Camembert example" introduced in Gauthier et al. (1986) to demonstrate the challenge of velocity estimation with FWI. We also display components of $\boldsymbol{U}(\boldsymbol{x})$ and $\boldsymbol{V}(\boldsymbol{x})$ for the Camembert example, to illustrate the properties of the projection basis discussed in the "Theory" section.

All of the numerical results are for the source pulse

$$f(t) = \cos(\omega_o t)\exp\left[-\frac{(2\pi B)^2 t^2}{2}\right], \tag{54}$$

with central frequency $\omega_o/(2\pi) = 6$ Hz and bandwidth $B = 4$ Hz. For details on the numerically simulated data, see Appendix A. To choose τ, we use $\omega_o/(2\pi) + B = 10$ Hz as the 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.

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

Topography of the objective function

Consider the velocity model shown in Figure 2a, 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, whereas the bottom region has the faster velocity $c_b = 3000$ m/s. The purpose of this example is to visualize the objective function, so we do not run Algorithm 2 and we do not use a search velocity of the form given in equation 46. Instead, we sweep a two-parameter search space: The first parameter is the interface position in the search interval $[0.47\ \text{km}, 1.95\ \text{km}]$, measured as the depth of the leftmost point of the interface. The actual position is 1.2 km. The second parameter is the contrast c_b/c_t in the interval $[1, 3]$. The actual contrast is two. The angle of the interface is kept constant and equal to the actual angle.

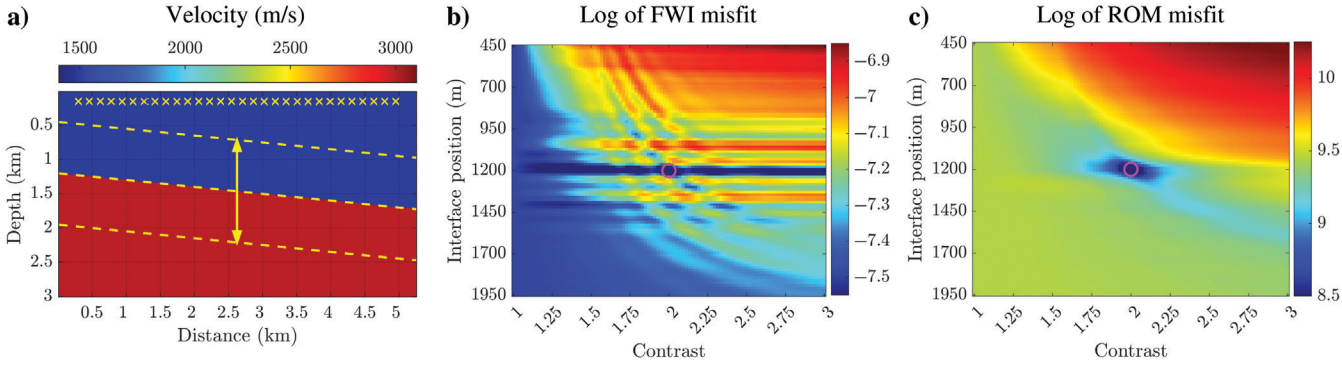

Figure 2. Objective functions topography study: (a) velocity model used in objective topography study. The dashed middle line shows the actual interface location, whereas the dashed top and bottom lines show the extent of the interface location parameter sweep. All $N_s = 30$ colocated sources/receivers are shown as yellow ×. Velocity colorbar is in m/s. (b and c) Decimal logarithms of the objective functions 55 and 56 versus the interface position and velocity contrast. The actual position and contrast parameters are indicated by the magenta circle. These true values are not included in the search space.

In Figure 2b and 2c, we display the decimal logarithms of two objective functions, calculated for $N_s = 30$ colocated sources/receivers and $N_t = 39$ time samples at interval $\tau = 0.0435$ s. The first objective function is for the FWI approach,

$$\mathcal{O}^{\mathrm{FWI}}(v) = \sum_{k=0}^{2N_t-1} \|\mathrm{Triu}(\boldsymbol{D}_k(v) - \boldsymbol{D}_k)\|_2^2, \tag{55}$$

where $\boldsymbol{D}_k(v)$ is the $N_s \times N_s$ data matrices for the search velocity $v(\boldsymbol{x})$ and $\mathrm{Triu}: \mathbb{R}^{N_s \times N_s} \mapsto \mathbb{R}^{N_s(N_s+1)/2}$ is the mapping that takes a symmetric $N_s \times N_s$ matrix, extracts its upper triangular part, including the main diagonal, and arranges its entries into a $N_s(N_s+1)/2$-dimensional column vector. The second objective function measures the misfit of the ROM:

$$\mathcal{O}^{\mathrm{ROM}}(v) = \|\mathrm{Triu}(\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v) - \boldsymbol{\mathcal{A}}^{\mathrm{ROM}})\|_2^2. \tag{56}$$

This corresponds to the particular case $d = k = N_t$ of the objective function in equation 49.

We observe in Figure 2b that the FWI objective function displays numerous local minima, at points in the search space that are far from the true one, marked in the plots with the magenta circle. There is no minimum at this circle because the exact values of the interface position and contrast are not in our parameter grid search space. The clearly visible horizontal stripes shown in Figure 2b are manifestations of cycle skipping. The ROM operator misfit shown in Figure 2c is smooth and has a single minimum, at the true interface position and contrast.

The "Camembert" example

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

The search space $\mathcal{C}$ has dimension $N = 20 \times 20 = 400$, and the velocity is parameterized as shown in equation 46, with the constant initial guess $c_o(\boldsymbol{x}) = \bar{c} = 3000$ m/s and the Gaussian basis functions,

$$\phi_l(\boldsymbol{x}) = \frac{1}{2\pi\sigma_\phi\sigma_\phi^\perp} \exp\left[-\frac{(x^\perp - x_l^\perp)^2}{2(\sigma_\phi^\perp)^2} - \frac{(x - x_l)^2}{2\sigma_\phi^2}\right], \tag{57}$$

with standard deviation $\sigma_\phi^\perp = 55.5$ m in the horizontal (distance) direction and $\sigma_\phi = 69.4$ m in depth. Here, we use the system of coordinates $\boldsymbol{x} = (x^\perp, x)$, with depth coordinate x and distance coordinate $x^\perp$ orthogonal to it. 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 $\Omega_{\mathrm{im}} = [95\ \mathrm{m}, 1905\ \mathrm{m}] \times [119\ \mathrm{m}, 2381\ \mathrm{m}] \subset \Omega$. Note that $2\sigma_\phi$ and $2\sigma_\phi^\perp$ are smaller than half the wavelength $\bar{c}/(10\ \mathrm{Hz}) = 300$ m corresponding to the essential Nyquist frequency. Hence, the velocity is overparameterized and we stabilize the inversion with the adaptive Tikhonov regularization described in Appendix D.

We show in Figure 4a–4d the velocity estimates obtained with Algorithm 2, implemented with $\ell = 9$, the number of iterations per layer $n_{\mathrm{iter}} = 4$, and with the restriction parameter $d = N_t$. The plots in Figure 4e–4h are the velocity estimates obtained with the FWI approach, which minimizes the objective function,

$$\mathcal{L}_i^{\mathrm{FWI}}(\boldsymbol{\eta}) = \mathcal{O}^{\mathrm{FWI}}(v(\cdot;\boldsymbol{\eta})) + \mu_i^{\mathrm{FWI}}\|\boldsymbol{\eta}\|_2^2, \tag{58}$$

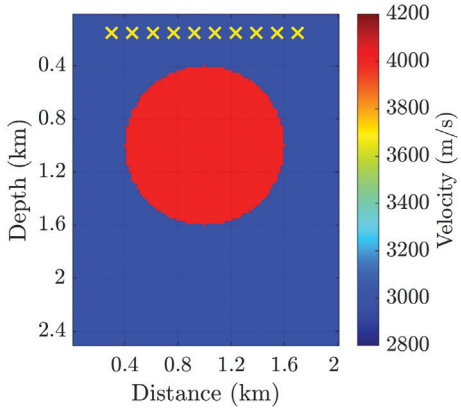

Figure 3. Camembert velocity model. All $N_s = 10$ colocated sources/receivers are shown as yellow ×. Velocity colorbar is in m/s.

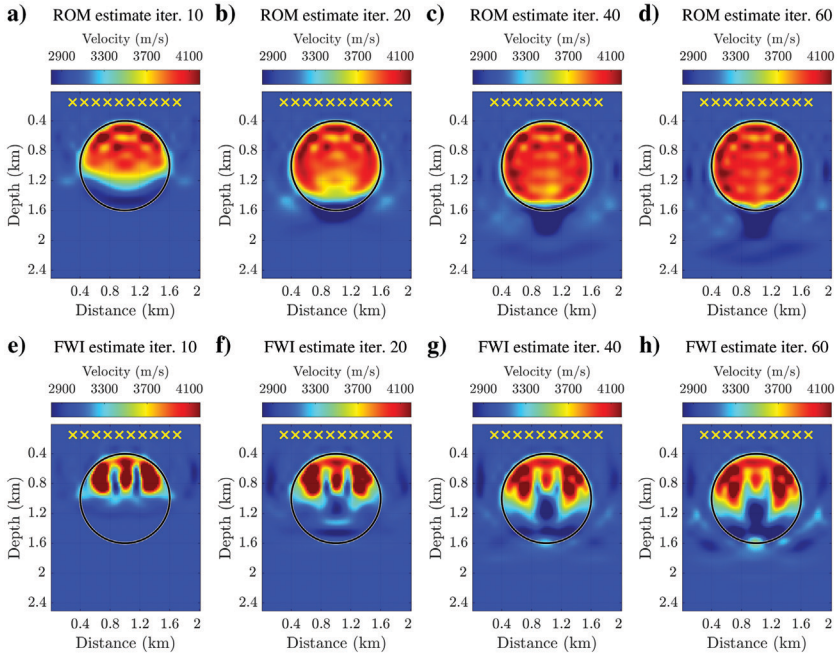

Figure 4. Estimated velocity after 10–60 Gauss-Newton iterations: (a–d) ROM-based velocity estimates and (e–h) FWI velocity estimates. The true inclusion boundary is shown as a black circle. All $N_s = 10$ colocated sources/receivers are shown as yellow ×. Velocity colorbars are in m/s and all plots share the same color scale.

with the same time windowing of the data as in the ROM-based estimation. The Tikhonov regularization parameter μ_i^{FWI} is computed as explained in Appendix D.

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

Illustration of the orthonormal basis

We display in Figure 5a the snapshot $u^{(s)}(4\tau, \boldsymbol{x})$ in the medium with the Camembert inclusion and in Figure 5c the snapshot computed with the reference, constant velocity $\bar{c} = 3000$ m/s. The source is in the middle of the array, indicated in the plots by the circle, and indexed by $s = 5$. Obviously, the snapshot in the true medium is different from the one in the reference medium. In the reference medium, the wave is a spherical wave emitted by the point source and reflected by the top surface modeled as a sound soft boundary. In the true Camembert model medium, the wave is scattered at the boundary and at the top of the inclusion, and it travels further down for the same $t = 4\tau$, due to the fast inclusion.

The corresponding components of the orthonormal basis stored in $\mathbf{V}(\boldsymbol{x})$, called $\mathrm{v}^{(5)}(4\tau, \boldsymbol{x})$, are shown in Figure 5b and 5d. They illustrate the second and third attributes of the orthonormal basis, stated in the outline of our velocity estimation method. Indeed, the basis function in the true and reference medium is very similar. They both have a localized peak near the deepest point reached by the wave at the instance $t = 4\tau$ and they are oscillatory away from it. The scattering at the top of the inclusion does not have a strong effect on the basis function, but the kinematics makes a difference. As mentioned previously, the wave penetration at $t = 4\tau$ is deeper in the true medium, due to the fast inclusion, so the localized peaks are in different locations.

VELOCITY ESTIMATION WITH NOISY AND TOWED-STREAMER DATA

In this section, we present velocity estimation results with noisy measurements and with the array response matrix $\boldsymbol{\mathcal{M}}(t)$ assembled from towed-streamer type measurements. In both cases, we have an uncertainty of the data, which affects the computation of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$. There are two critical steps in Algorithm 1 that must be addressed, and they both involve the mass matrix $\boldsymbol{M}$ computed at step 3, which will likely be neither symmetric nor positive definite. These properties are needed for the computation of the Cholesky square root $\boldsymbol{R}$ at step 4 and the inverse $\boldsymbol{R}^{-1}$ that gives the output of the algorithm. The lack of symmetry is easy to fix, but to ensure the positive definiteness, we need a regularization procedure that involves a spectral projection of $\boldsymbol{M}$ on the space of its leading eigenvectors, corresponding to the significant eigenvalues. These eigenvectors and eigenvalues are least affected by the uncertainty. The regularization procedure is not straightforward because we must preserve the causality of $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ in order for the velocity estimation to succeed. We explain it in detail in Appendix E.

To assemble the matrix $\boldsymbol{\mathcal{M}}(t)$ from towed-streamer measurements, we use source-receiver reciprocity on-the-fly to fill in the missing off-diagonal entries in $\boldsymbol{\mathcal{M}}(t)$. To compute the diagonal entries, corresponding to the source being also a receiver, we use interpolation of the values at nearby measurement locations, two on the left and two on the right. We use Lagrange polynomial interpolation in the Fourier (frequency) domain, for

$$\int_{\mathbb{R}} dt\, e^{i\omega t}[\boldsymbol{\mathcal{M}}(t) - \mathcal{F}[\bar{c}](t)]. \tag{59}$$

Then, we inverse the Fourier transform to obtain $\boldsymbol{\mathcal{M}}(t)$.

Numerical results

We do not show the Camembert estimation for uncertain measurements because the information needed to get the good result in Figure 4 requires accurate knowledge of $\boldsymbol{\mathcal{M}}(t)$. This is not the fault of the inversion method. It is due to the fact that the bottom part of the Camembert inclusion gives a very weak signal at the array, which is accounted for in the small eigenvalues of the mass matrix. Any uncertainty of the data will perturb significantly these eigenvalues and the associated eigenvectors, so the ROM inversion is not better than that with FWI.

We present instead velocity estimation results for a section of the Marmousi model shown in Figure 6a, where we exclude the portion of the water down to a depth of 266 m. The domain is $\Omega = [0, 5.25\ \text{km}] \times [0, 3\ \text{km}]$. The data sampling for the ROM construction is $\tau = 0.0435$ s and the number of snapshots that span the approximation space is $N_t = 40$. The colocated sources/receivers are located underwater at a depth of 150 m and they emit the same pulse given in equation 54. We present results in two settings. First, when working with noisy data, we use an array of $N_s = 30$ colocated sources/receivers, separated by a distance of 166.66 m. Second, when working with data approximated from towed-streamer type measurements, we use closely spaced receivers, at 16.66 m apart, to carry out the interpolation of the measurements and fill in the missing zero offset data. Then, we subsample the result before we input it in Algorithm 1, by keeping $N_s = 40$ sources/receivers separated by a distance of 116.66 m.

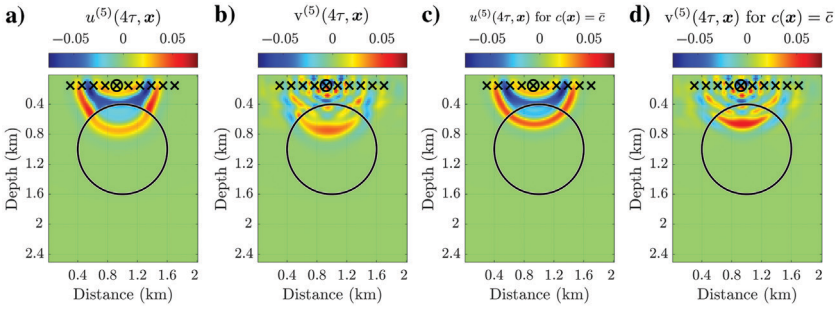

Figure 5. Wavefield snapshots and orthonormal basis components at time instance $t = 4\tau$, corresponding to the center left source, indexed by $s = 5$, shown as a black ◯: (a and b) for the true velocity $c(\boldsymbol{x})$ shown in Figure 3 and (c and d) for the reference medium, with $c(\boldsymbol{x}) \equiv \bar{c} = 3000$ m/s. The $N_s = 10$ colocated sources/receivers are shown as black ×. All of the plots share the same color scale.

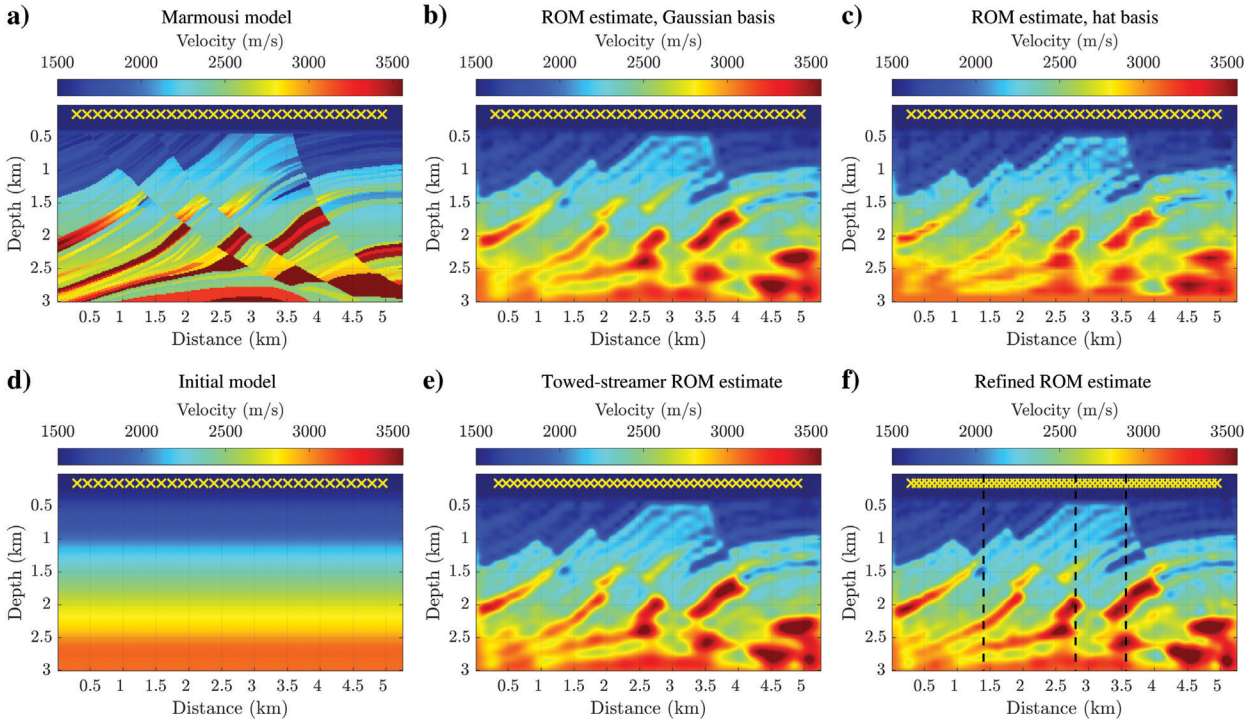

Figure 6. The ROM-based velocity estimates for the Marmousi model with noisy data and towed-streamer measurements: (a) the section of the Marmousi model, (b) velocity estimate from noisy data with Gaussian basis functions parameterization of v, (c) velocity estimate from noisy data with hat basis functions parameterization of v, (d) initial guess model $c_o(\boldsymbol{x})$, (e) velocity estimate from towed-streamer measurements, and (f) velocity estimate refinement from data gathered on dense array sensors and at a small time interval τ. All the sources/receivers, $N_s = 30$ in (a–d), $N_s = 40$ in (e), and $N_s = 60$ in (f), are shown as yellow ×. Velocity colorbars are in m/s. All plots share the same color scale.

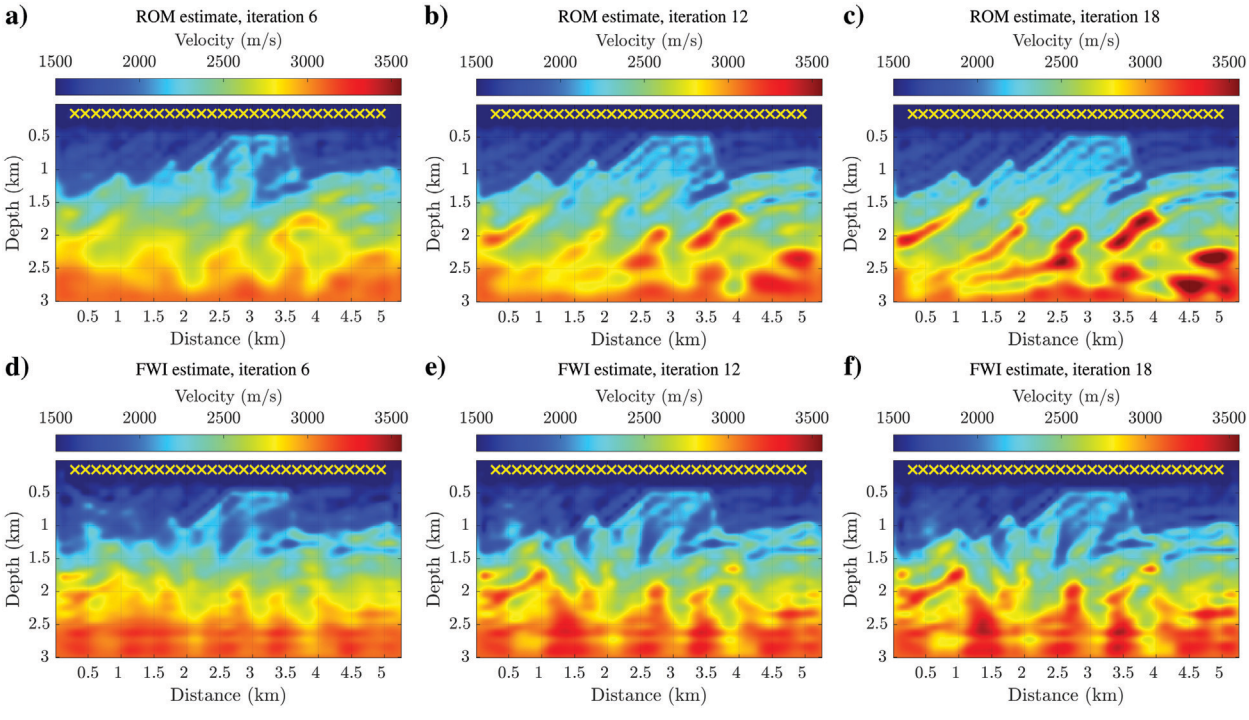

Figure 7. Velocity estimates for the Marmousi model with noisy data after 6, 12, and 18 Gauss-Newton iterations: (a–c) ROM-based approach and (d–f) FWI approach. The $N_s = 30$ colocated sources/receivers are shown as yellow ×. Velocity colorbars are in m/s and all plots share the same color scale.

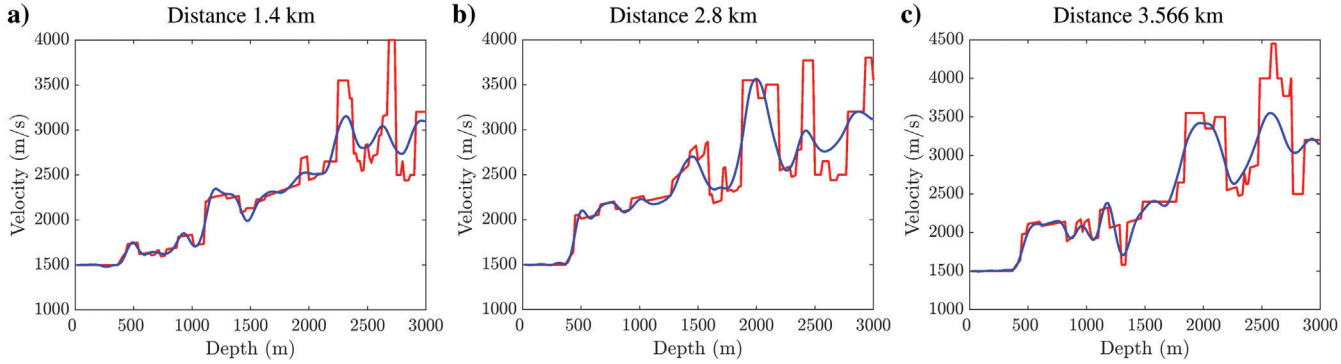

Figure 8. Vertical slices of the Marmousi model velocity (the red lines) and its refined ROM estimate (the blue lines) at distances shown as the dashed lines in Figure 6f.

In Figure 7a–7c, we show the ROM-based inversion results obtained from data contaminated with 1% additive noise described in Appendix A. We used $\ell = 6$ layers in Algorithm 2, with $n_{\text{iter}} = 3$ iterations per layer, and the restriction parameter $d = 10$. The ROM operator is regularized as explained in Appendix E with the spectral threshold parameter set to $r = N_t - 9 = 31$. The velocity is parameterized as given in equation 46, with the initial guess $c_o(\boldsymbol{x})$ displayed in Figure 6d. We used $N = 50 \times 30 = 1500$ Gaussian basis functions defined as in equation 57, 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 $\Omega_{\text{im}} = [103\text{ m}, 5147\text{ m}] \times [97\text{ m}, 2903\text{ m}]$ contained in Ω. Figure 7d–7f shows the FWI results computed for noiseless data. We use the same parameterization of the search velocity and invert in $\ell = 6$ layers with the same data windowing as in the ROM-based inversion.

We observe in Figure 7 that the ROM-based velocity estimation 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 duration of the measurements and the depth of the domain Ω, so that the artificial bottom boundary has no effect. 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 12th iteration and the result is far from the true model. Effectively, FWI is stuck in a local minimum.

In Figure 6b and 6c, we compare the ROM-based estimates obtained with two different choices of the basis functions in the parameterization from expression 46 of the search velocity. The Gaussian ones given in equation 57 and the commonly used piecewise linear hat functions, which interpolate between the values of zero and one on the same 50×30 inversion grid. The estimate with the Gaussian basis looks smoother, as expected, but the point of this comparison is to illustrate that the inversion is very mildly sensitive to the parameterization of the search velocity, once the inversion grid is fixed.

We show in Figure 6f how the velocity estimation improves if we double the number of colocated sources/receivers to $N_s = 60$, decrease the time sampling to $\tau = 0.0333$ s and increase N_t to 50, while also setting $r = N_t - 17 = 33$. The inversion is carried out as mentioned previously, except that the parameterization 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. We use the estimate from Figure 6b as an initial guess. Because this initial velocity estimate is already very good, it is sufficient to perform $n_{\text{iter}} = 4$ 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.

To illustrate better the quality of the refined ROM estimate in Figure 6f, we display in Figure 8 the true and refined estimated velocity for three vertical slices, at distances of 1.4, 2.8, and 3.566 km. 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 previously.

We end the section with the velocity estimate obtained with the array response matrix estimated from towed-streamer type measurements, which is shown in Figure 6e. We observe that this estimate is practically the same as the one in Figure 6b.

CONCLUSION

We introduced a novel approach for velocity estimation based on a ROM of the wave operator. The ROM is computed from the data gathered by an array of colocated sources and receivers. Such data can be approximated in geophysics applications from towed-streamer type measurements. No prior information on the medium is used, except for the assumption that the velocity is known in the immediate vicinity of the sensors. Although 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 wavefield at uniformly spaced time steps. This space is not 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 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 FWI. In particular, for a low-dimensional velocity model where we can plot the objective functions, we obtain that the ROM misfit objective function has convexity properties, whereas 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.

Our ROM construction uses that the data matrices have a symmetric inner-product mathematical expression. This requirement is

the main impediment to having a straightforward extension of the methodology to more general data acquisition setups, with sources and receivers at very different locations. We hope to address this open challenge in future research.

ACKNOWLEDGMENTS

We thank the associate editor and the anonymous reviewers for the valuable feedback on how to improve the exposition. 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.

DATA AND MATERIALS AVAILABILITY

No data have been required for this paper.

APPENDIX A

NUMERICALLY SIMULATED DATA

The data for the numerical experiments are computed with a time-domain wave equation solver for equations 1 and 2, with Laplacian 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 $\tau_{\rm f} = \tau/20$. Using equation 13, we obtain the finely sampled data $\boldsymbol{D}_k^{\rm f}$, for $k = 0, 1, \ldots, n_{\rm f}$, where $n_{\rm f} = 20(2N_{\rm t} - 1)$.

The noisy data are computed as follows. Define

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

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

We now explain how we compute the second derivative data matrices. We begin by extending the finely sampled data evenly in discrete time to obtain $\boldsymbol{D}_j^{\rm fe}$, $j = -n_{\rm f}, \ldots, n_{\rm f}$, with $\boldsymbol{D}_k^{\rm f} = \boldsymbol{D}_{\pm k}^{\rm fe}$, $k = 0, 1, \ldots, n_{\rm f}$. Then, we take the discrete Fourier transform of $(\boldsymbol{D}_j^{\rm fe})_{j=-n_{\rm f}}^{n_{\rm f}}$ and differentiate it 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 $\ddot{\boldsymbol{D}}_j^{\rm fe}$, at $j = -n_{\rm f}, \ldots, n_{\rm f}$, the finely sampled second derivative data. Finally, we subsample $\boldsymbol{D}_j^{\rm fe}$ and $\ddot{\boldsymbol{D}}_j^{\rm fe}$ to obtain

$$\boldsymbol{D}_k = \boldsymbol{D}_{20k}^{\rm fe}, \quad \ddot{\boldsymbol{D}}_k = \ddot{\boldsymbol{D}}_{20k}^{\rm fe}, \quad k = 0, 1, \ldots, 2N_{\rm t} - 1. \tag{A-2}$$

APPENDIX B

CAUSAL CONSTRUCTION OF THE ROM

Here, we prove that the upper left $kN_{\rm s} \times kN_{\rm s}$ block of $\boldsymbol{\mathcal{A}}^{\rm ROM}$, denoted by $[\boldsymbol{\mathcal{A}}^{\rm ROM}]_k$, 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 $k = 1, \ldots, N_{\rm t}$.

Let us begin by writing $[\boldsymbol{\mathcal{A}}^{\rm ROM}]_k$ from equation 26

$$\begin{aligned}[\boldsymbol{\mathcal{A}}^{\rm ROM}]_k &= \begin{pmatrix} \mathbf{I}_{kN_{\rm s}} & \mathbf{0} \end{pmatrix} \boldsymbol{R}^{-T}\boldsymbol{S}\boldsymbol{R}^{-1} \begin{pmatrix} \boldsymbol{I}_{kN_{\rm s}} \\ \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} [\boldsymbol{R}]_k^{-T} & \mathbf{0} \end{pmatrix} \boldsymbol{S} \begin{pmatrix} [\boldsymbol{R}]_k^{-1} \\ \mathbf{0} \end{pmatrix} \\ &= [\boldsymbol{R}]_k^{-T}[\boldsymbol{S}]_k[\boldsymbol{R}]_k^{-1}, \end{aligned} \tag{B-1}$$

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

At step 3, Algorithm 1 computes from $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k-2}$ the upper left $kN_{\rm s} \times kN_{\rm s}$ block of $\boldsymbol{M}$, denoted by $[\boldsymbol{M}]_k$, and also $[\boldsymbol{S}]_k$. The Cholesky factorization in equation 22 and the block upper triangular structure of $\boldsymbol{R}$ give

$$[\boldsymbol{M}]_k = \begin{pmatrix} \boldsymbol{I}_{kN_{\rm s}} & \mathbf{0} \end{pmatrix} \boldsymbol{R}^T\boldsymbol{R} \begin{pmatrix} \boldsymbol{I}_{kN_{\rm s}} \\ \mathbf{0} \end{pmatrix} = [\boldsymbol{R}]_k^T[\boldsymbol{R}]_k. \tag{B-2}$$

This shows that $[\boldsymbol{R}]_k$ is the Cholesky square root of $[\boldsymbol{M}]_k$, computed in Algorithm 1. The result follows from equation B-1.

APPENDIX C

ALGEBRAIC STRUCTURE OF THE ROM

We explain here that the entries of the ROM operator $\boldsymbol{\mathcal{A}}^{\rm ROM}$ decay away from the main diagonal, which is why we can use the restriction mapping $\text{Rest}_{d,k}$ defined in equation 47 to reduce the computational cost of the inversion. Let us write

$$\boldsymbol{V}(\boldsymbol{x}) = (\mathbf{v}_0(\boldsymbol{x}), \ldots, \mathbf{v}_{N_{\rm t}-1}(\boldsymbol{x})), \tag{C-1}$$

where $\mathbf{v}_j(\boldsymbol{x}) \in \mathbb{R}^{1\times N_{\rm s}}$, for $j = 0, \ldots, N_{\rm t} - 1$. We obtain from equation 8 that the $N_{\rm s} \times N_{\rm s}$ blocks of $\boldsymbol{\mathcal{A}}^{\rm ROM}$ are

$$\boldsymbol{\mathcal{A}}_{i,j}^{\rm ROM} = \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\mathbf{x})\boldsymbol{\mathcal{A}}\mathbf{v}_j(\boldsymbol{x}), \quad i,j = 0,\ldots,N_{\rm t}-1. \tag{C-2}$$

Moreover, the Gram-Schmidt orthogonalization from equation 27 gives

$$\boldsymbol{u}_j(\boldsymbol{x}) = \sum_{q=0}^{j} \mathbf{v}_q(\boldsymbol{x})\boldsymbol{R}_{q,j}, \tag{C-3}$$

and conversely

$$\mathbf{v}_j(\boldsymbol{x}) = \sum_{q=0}^{j} \boldsymbol{u}_q(\boldsymbol{x})\boldsymbol{\Gamma}_{q,j}, \tag{C-4}$$

where

$$\boldsymbol{\Gamma} = \boldsymbol{R}^{-1} = \begin{pmatrix} \boldsymbol{\Gamma}_{0,0} & \boldsymbol{\Gamma}_{0,1} & \cdots & \boldsymbol{\Gamma}_{0,N_t-1} \\ \boldsymbol{0} & \boldsymbol{\Gamma}_{1,1} & \cdots & \boldsymbol{\Gamma}_{1,N_t-1} \\ \vdots & \vdots & \vdots & \\ & & & \boldsymbol{\Gamma}_{N_t-1,N_t-1} \end{pmatrix} \tag{C-5}$$

is block upper triangular, like $\boldsymbol{R}$.

Now let us substitute equation C-4 into equation C-2, to obtain

$$\begin{aligned} \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}_{i,j} &= \sum_{q=0}^{j} \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\boldsymbol{x}) \mathcal{A}\boldsymbol{u}_q(\boldsymbol{x}) \boldsymbol{\Gamma}_{q,j} \\ &= -\sum_{q=0}^{j} \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\boldsymbol{x}) \partial_t^2 \boldsymbol{u}(q\tau,\boldsymbol{x}) \boldsymbol{\Gamma}_{q,j}. \end{aligned} \tag{C-6}$$

We use next the Whittaker-Shannon interpolation formula, which says that if τ satisfies the Nyquist criterion, then

$$u(t,\boldsymbol{x}) = \sum_{s=-\infty}^{\infty} u_{|s|}(\boldsymbol{x}) \mathrm{sinc}\left[\frac{\pi(t-s\tau)}{\tau}\right]. \tag{C-7}$$

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

$$\tau^2 \partial_t^2 u(q\tau,\boldsymbol{x}) = \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} u_{|q-s|}(\boldsymbol{x}) - \frac{\pi^2}{3} u_q(\boldsymbol{x}), \tag{C-8}$$

and substituting into equation C-6, we obtain

$$\begin{aligned} \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}_{i,j} &= \frac{1}{\tau^2} \sum_{q=0}^{j} \boldsymbol{\Gamma}_{q,j} \Bigg\{ \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\boldsymbol{x}) \boldsymbol{u}_q(\boldsymbol{x}) \\ &\quad - \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\boldsymbol{x}) \boldsymbol{u}_{|q-s|}(\boldsymbol{x}) \Bigg\} \\ &= \frac{1}{\tau^2} \sum_{q=0}^{j} \boldsymbol{\Gamma}_{q,j} \left\{ \boldsymbol{R}_{i,q} - \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} \boldsymbol{R}_{i,|q-s|} \right\}. \end{aligned} \tag{C-9}$$

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

Because $\boldsymbol{\Gamma}_{q,j} = 0$ for $q > j$, and $\boldsymbol{R}_{i,q} = 0$ for $i > q$, the first term on the right side of equation C-9 is zero for $i > j$. However, we are interested only in the block upper triangular part of $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ (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 equation C-9. Each term in this series adds an sth diagonal, whose entries decay as $1/s^2$. Thus, only the first few block diagonals are large.

APPENDIX D

IMPLEMENTATION OF THE INVERSION

In principle, the optimization at step 3 of Algorithm 2 could have a constraint on $\boldsymbol{\eta}$ to ensure that the search velocity in equation 46 is positive. We did not need such a constraint in our numerical simulations, as the velocity has stayed positive throughout the iterations.

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

$$\mathcal{L}_i^{\mathrm{reg}}(\boldsymbol{\eta}) = \mu_i \|\boldsymbol{\eta}\|_2^2, \tag{D-1}$$

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

$$\mathcal{E}(\boldsymbol{\eta}; d, k_l) = \mathrm{Rest}_{d,k_l}([\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v(\cdot;\boldsymbol{\eta})) - \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}]_{k_l}) \tag{D-2}$$

be the $N_s d(2k-d+1)/2$-dimensional residual vector, whose Euclidean norm squared appears in equation 49. The Jacobian of the objective function evaluated at $\boldsymbol{\eta} = \boldsymbol{\eta}^{(i-1)}$ is the matrix

$$\boldsymbol{J}^{(i)} = \nabla_{\boldsymbol{\eta}} \mathcal{E}(\boldsymbol{\eta}^{(i-1)}; d, k_l) \in \mathbb{R}^{dN_s(kN_s - (dN_s-1)/2) \times N}. \tag{D-3}$$

We always choose the parameterization of the velocity from equation 46 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

$$\mu_i = (\sigma^{(i)}_{\lfloor \gamma N \rfloor})^2. \tag{D-4}$$

The choice of γ depends on the parameterization in equation 46. Because it is not clear what is the resolution of the inversion, we choose to overparameterize the velocity, and stabilize the inversion with a small γ, in the range $(0.2, 0.4)$. For the results presented in the paper, we used $\gamma = 0.25$.

The Gauss-Newton update direction for the objective function in equation 50, regularized with $\mathcal{L}_i^{\mathrm{reg}}(\boldsymbol{\eta})$ from equation D-1, is

$$\boldsymbol{d}^{(i)} = -((\boldsymbol{J}^{(i)})^T \boldsymbol{J}^{(i)} + \mu_i \boldsymbol{I}_N)^{-1} (\boldsymbol{J}^{(i)})^T \boldsymbol{r}^{(i)}, \tag{D-5}$$

where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix and $\boldsymbol{r}^{(i)}$ is the residual vector in equation D-2 evaluated at $\boldsymbol{\eta}^{(i-1)}$. Note that equation D-5 is the same as the Levenberg-Marquardt update direction for equation D-2 with damping μ_i.

Given the update direction $\boldsymbol{d}^{(i)}$, we use a line search,

$$\alpha^{(i)} = \underset{\alpha \in (0, \alpha_{\max})}{\mathrm{argmin}}\, \mathcal{L}_i(\boldsymbol{\eta}^{(i-1)} + \alpha \boldsymbol{d}^{(i)}), \tag{D-6}$$

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

$$\boldsymbol{\eta}^{(i)} = \boldsymbol{\eta}^{(i-1)} + \alpha^{(i)} \boldsymbol{d}^{(i)}. \tag{D-7}$$

We use a similar regularization strategy for the FWI objective function in equation 58: If we let $\mathcal{E}^{\mathrm{FWI}}(\boldsymbol{\eta})$ be the $N_t N_s(N_s+1)$-dimensional residual vector, with entries

$$(\mathcal{E}_j^{\mathrm{FWI}}(\boldsymbol{\eta}))_{j=kN_s(N_s+1)/2+1}^{(k+1)N_s(N_s+1)/2} = \mathrm{Triu}(\boldsymbol{D}_k(v) - \boldsymbol{D}_k), \tag{D-8}$$

for $k = 0, \ldots, 2N_t - 1$, then its Jacobian evaluated at $\boldsymbol{\eta} = \boldsymbol{\eta}^{(i-1)}$ is

$$\boldsymbol{J}^{\mathrm{FWI},(i)} = \nabla_{\boldsymbol{\eta}} \mathcal{R}^{\mathrm{FWI}}(\boldsymbol{\eta}^{(i-1)}) \in \mathbb{R}^{N_t N_s(N_s+1) \times N}, \tag{D-9}$$

where we assume $N \le N_t N_s(N_s+1)$. Furthermore, for the same fixed parameter γ used in the ROM approach, we set $\mu_i^{\text{FWI}} = (\sigma_{\lfloor \gamma N \rfloor}^{\text{FWI},(i)})^2$, where $\{\sigma_j^{\text{FWI},(i)}\}_{j=1}^N$ is the singular value of $\boldsymbol{J}^{\text{FWI},(i)}$, sorted in decreasing order.

APPENDIX E

REGULARIZATION OF THE ROM

Let us denote by $\{\boldsymbol{D}_j^{\text{N}}\}_{j=0}^{2N_t-1}$ the uncertain data matrices that are either contaminated with noise (see Appendix A) or approximated from the towed-streamer measurements. Source-receiver reciprocity is built into the approximation for towed-streamer measurements, but it does not hold for noisy array data. To ensure symmetry, we transform $\boldsymbol{D}_j^{\text{N}}$ into $(1/2)(\boldsymbol{D}_j^{\text{N}} + (\boldsymbol{D}_j^{\text{N}})^T)$.

The mass and stiffness matrices computed at step 3 of Algorithm 1 are denoted by $\boldsymbol{M}^{\text{N}}$ and $\boldsymbol{S}^{\text{N}}$. In theory, they should be positive definite matrices, but they will have several 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

$$\boldsymbol{M}^{\text{N}} = \boldsymbol{Z}^{\text{N}} \boldsymbol{\Lambda}^{\text{N}} (\boldsymbol{Z}^{\text{N}})^T \tag{E-1}$$

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

$$\boldsymbol{Z}^{\text{N},r} = (Z_{jl}^{\text{N}})_{1\le j\le N_t N_s, 1\le l\le rN_s} \in \mathbb{R}^{N_t N_s \times rN_s}, \tag{E-2}$$

to define the projected mass matrix

$$\boldsymbol{\Lambda}^{\text{N},r} = (\boldsymbol{Z}^{\text{N},r})^T \boldsymbol{M}^{\text{N}} \boldsymbol{Z}^{\text{N},r} = \text{diag}(\lambda_1^{\text{N}}, \ldots, \lambda_{rN_s}^{\text{N}}). \tag{E-3}$$

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 43). Thus, we need an additional transformation to recover causality. 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 obtain the transformation using another ROM, for the propagator operator (Borcea et al., 2018, 2021),

$$\mathcal{P} = \cos\left(\tau\sqrt{\mathcal{A}}\right). \tag{E-4}$$

The ROM propagator is obtained from the Galerkin approximation of the time stepping equation:

$$\boldsymbol{u}_{j+1}(\boldsymbol{x}) = 2\mathcal{P}\boldsymbol{u}_j(\boldsymbol{x}) - \boldsymbol{u}_{|j-1|}(\boldsymbol{x}), \quad j \ge 0, \tag{E-5}$$

obtained from equation 39 evaluated at $t = j\tau$ and $\Delta t = \tau$, for $j \ge 0$. The approximation space is the same as in the computation of $\mathcal{A}^{\text{ROM}}$, i.e., range$(\boldsymbol{U}(\boldsymbol{x}))$. If we let $\tilde{\boldsymbol{u}}_{G,j} = \boldsymbol{U}(\boldsymbol{x})\tilde{\boldsymbol{g}}_j$ be the Galerkin approximation at instance $t = j\tau$, this satisfies the algebraic system of equations

$$\underbrace{\int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x})}_{\boldsymbol{M}}(\tilde{\boldsymbol{g}}_{j+1} + \tilde{\boldsymbol{g}}_{|j-1|}) = 2\underbrace{\int_\Omega d\boldsymbol{x}\, \boldsymbol{U}^T(\boldsymbol{x})\mathcal{P}\boldsymbol{U}(\boldsymbol{x})}_{\tilde{\boldsymbol{S}}}\tilde{\boldsymbol{g}}_j, \tag{E-6}$$

for $j \ge 0$. Note how the same data-driven mass matrix $\boldsymbol{M}$ appears in this equation. The propagator stiffness matrix $\tilde{\boldsymbol{S}}$ also is data driven, with $N_s \times N_s$ blocks given by

$$\begin{aligned}\tilde{\boldsymbol{S}}_{i,j} &= \left\langle \boldsymbol{u}_i, \cos\left(\tau\sqrt{\mathcal{A}}\right)\boldsymbol{u}_j \right\rangle \\ &= \frac{1}{2}\langle \boldsymbol{u}_i, \boldsymbol{u}_{j+1} + \boldsymbol{u}_{|j-1|}\rangle \\ &= \frac{1}{4}(\boldsymbol{D}_{i+j+1} + \boldsymbol{D}_{|i-j-1|} + \boldsymbol{D}_{|i+j-1|} + \boldsymbol{D}_{|i-j+1|}),\end{aligned} \tag{E-7}$$

where $0 \le i, j \le N_t - 1$. Using the same block Cholesky factorization of $\boldsymbol{M}$ from equation 22 and multiplying the preceding Galerkin equation by $\boldsymbol{R}^{-1}$, we obtain the time stepping scheme in the ROM space

$$\tilde{\boldsymbol{u}}_{j+1}^{\text{ROM}} = 2\mathcal{P}^{\text{ROM}}\tilde{\boldsymbol{u}}_j^{\text{ROM}} - \tilde{\boldsymbol{u}}_{|j-1|}^{\text{ROM}}, \tag{E-8}$$

the algebraic analog of equation E-5, where

$$\tilde{\boldsymbol{u}}_j^{\text{ROM}} = \boldsymbol{R}\tilde{\boldsymbol{g}}_j, \quad j \ge 0, \tag{E-9}$$

are the ROM snapshots and

$$\mathcal{P}^{\text{ROM}} = \boldsymbol{R}^{-T}\tilde{\boldsymbol{S}}\boldsymbol{R}^{-1} = \int_\Omega d\boldsymbol{x}\boldsymbol{V}^T(\boldsymbol{x})\mathcal{P}\boldsymbol{V}(\boldsymbol{x}) \tag{E-10}$$

is the ROM propagator. In the last equality, we used the definition of $\tilde{\boldsymbol{S}}$ and the Gram-Schmidt orthogonalization equation 27.

We refer the interested reader to Borcea et al. (2020) for a long and detailed analysis of $\mathcal{P}^{\text{ROM}}$. For our purpose, it suffices to say that it is useful to look at it because, as proved in Borcea et al. (2020, Appendix C), unlike $\mathcal{A}^{\text{ROM}}$, the matrix $\mathcal{P}^{\text{ROM}}$ is sparse, with a block tridiagonal structure. Moreover, $\mathcal{P}^{\text{ROM}}$ is determined by the same mass matrix as $\mathcal{A}^{\text{ROM}}$. Thus, even though the regularized matrix in equation E-3 is not in the right algebraic form, we can bring it into the right form by imposing the block tridiagonal structure of the resulting ROM propagator. To do this, we use the block-Lanczos algorithm (Golub and Van Loan, 2013, Chapter 10) that takes any symmetric matrix in $\mathbb{R}^{N_t N_s \times N_t N_s}$ and computes an orthogonal basis of $\mathbb{R}^{N_t N_s}$ that puts the matrix in block tridiagonal form.

We can now describe the regularization procedure. First, we compute the ROM propagator stiffness matrix $\tilde{\boldsymbol{S}}^{\text{N}}$, with blocks as given previously, in terms of the uncertain data $\{\boldsymbol{D}_j^{\text{N}}\}_{j=0}^{2N_t-1}$. Then, we project this matrix onto the range of $\boldsymbol{Z}^{\text{N},r}$, defined in equation E-2,

$$\tilde{\boldsymbol{S}}^{\text{N},r} = (\boldsymbol{Z}^{\text{N},r})^T \tilde{\boldsymbol{S}}^{\text{N}} \boldsymbol{Z}^{\text{N},r} \in \mathbb{R}^{rN_s \times rN_s}, \tag{E-11}$$

and we compute

$$\boldsymbol{P}^{\mathrm{N},r} = (\boldsymbol{\Lambda}^{\mathrm{N},r})^{-1/2}\tilde{\boldsymbol{S}}^{\mathrm{N},r}(\boldsymbol{\Lambda}^{\mathrm{N},r})^{-1/2} \in \mathbb{R}^{rN_\mathrm{s}\times rN_\mathrm{s}}. \tag{E-12}$$

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

$$\mathcal{P}^{\mathrm{ROM},r} = (\boldsymbol{Q}^{\mathrm{N},r})^T\boldsymbol{P}^{\mathrm{N},r}\boldsymbol{Q}^{\mathrm{N},r} \in \mathbb{R}^{rN_\mathrm{s}\times rN_\mathrm{s}} \tag{E-13}$$

is a block tridiagonal matrix with $N_\mathrm{s}\times N_\mathrm{s}$ blocks, which we call the regularized ROM propagator.

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

$$(\boldsymbol{Q}^{\mathrm{N},r})^T\boldsymbol{\Lambda}^{\mathrm{N},r}\boldsymbol{Q}^{\mathrm{N},r} = (\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{M}^{\mathrm{N}}\boldsymbol{\Pi}^{\mathrm{N},r} = (\boldsymbol{R}^{\mathrm{N},r})^T\boldsymbol{R}^{\mathrm{N},r}, \tag{E-14}$$

where

$$\boldsymbol{\Pi}^{\mathrm{N},r} = \boldsymbol{Z}^{\mathrm{N},r}\boldsymbol{Q}^{\mathrm{N},r} \in \mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times rN_\mathrm{s}}, \tag{E-15}$$

and $\boldsymbol{R}^{\mathrm{N},r} \in \mathbb{R}^{rN_\mathrm{s}\times rN_\mathrm{s}}$ is block upper triangular and well conditioned, due to the spectral truncation in equation E-3. Then, using the data-driven stiffness matrix $\boldsymbol{S}^{\mathrm{N}}$ computed at step 3 of Algorithm 1, we obtain the regularized operator ROM as

$$\mathcal{A}^{\mathrm{ROM},r} = (\boldsymbol{R}^{\mathrm{N},r})^{-T}(\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{S}^{\mathrm{N}}\boldsymbol{\Pi}^{\mathrm{N},r}(\boldsymbol{R}^{\mathrm{N},r})^{-1}. \tag{E-16}$$

Equation E-16 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 from equation E-15: Let $\boldsymbol{M}(v)$ and $\boldsymbol{S}(v)$ be the mass and stiffness matrices calculated 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

$$(\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{M}(v)\boldsymbol{\Pi}^{\mathrm{N},r} = \boldsymbol{R}^r(v)^T\boldsymbol{R}^r(v), \tag{E-17}$$

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

$$\mathcal{A}^{\mathrm{ROM},r}(v) = \boldsymbol{R}^r(v)^{-T}(\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{S}(v)\boldsymbol{\Pi}^{\mathrm{N},r}\boldsymbol{R}^r(v)^{-1}. \tag{E-18}$$

The velocity inversion is carried out as in Algorithm 2, with $\mathcal{A}^{\mathrm{ROM}}$ and $\mathcal{A}^{\mathrm{ROM}}(v)$ in equation 49 replaced by the regularized $\mathcal{A}^{\mathrm{ROM},r}$ and $\mathcal{A}^{\mathrm{ROM},r}(v)$. Note that the matrix with orthogonal columns $\boldsymbol{\Pi}^{\mathrm{N},r}$ used in equations E-17 and E-18 is computed once using the uncertain data and does not change over the course of velocity estimation.

We observe that due to the block algebra, even if we do not use a spectral truncation, i.e., set $r = N_\mathrm{t}$, the ROM operator in equation E-17 is not identical to the one computed with Algorithm 1. Nevertheless, they behave the same with respect to the inversion, as shown in Figure E-1, where we plot the logarithm of the objective function:

$$\mathcal{O}^{\mathrm{ROM},r}(v) = \|\mathrm{Triu}(\mathcal{A}^{\mathrm{ROM},r}(v) - \mathcal{A}^{\mathrm{ROM},r})\|_2^2, \tag{E-19}$$

for the same experiment as shown in Figure 2, for the cases $r = N_\mathrm{t}$ and $r = N_\mathrm{t} - 4$. There is little difference between Figure 2c and Figure E-1a and E-1b.

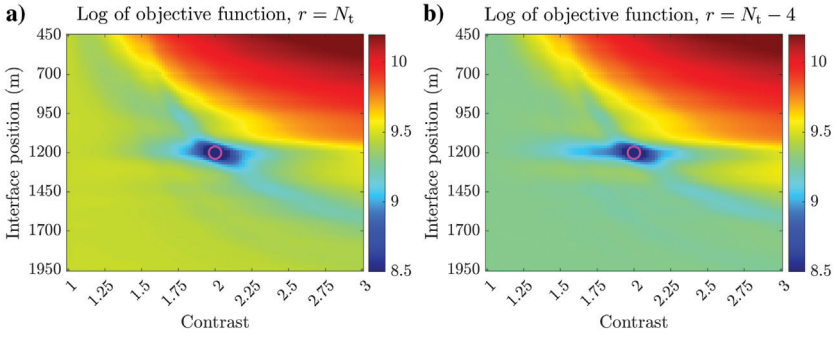

Figure E-1. Decimal logarithm of the objective function (equation E-19) versus the interface position and velocity contrast. The true parameters (shown in Figure 2) are indicated by the magenta circle.

APPENDIX F

THE REGULARIZATION THRESHOLD

Here, we explain how we choose the regularization threshold r for the ROM regularization procedure in equations E-2 and E-18. The idea is that r can be determined from the part of the spectrum of the mass matrix $\boldsymbol{M}^{\mathrm{N}}$ that is perturbed by the uncertainty. This can be estimated using the mass matrix $\boldsymbol{M}^{\mathrm{N}}(c_o)$ corresponding to the initial guess velocity $c_o(\boldsymbol{x})$, and perturbed in a similar way.

For noisy sensor array measurements, the matrices

$$\boldsymbol{E}_j^{\mathrm{N}} = \frac{1}{\sqrt{2}}(\boldsymbol{D}_j^{\mathrm{N}} - (\boldsymbol{D}_j^{\mathrm{N}})^T), \quad j = 0, \ldots, 2N_\mathrm{t} - 1, \tag{F-1}$$

can be considered as realizations of the additive noise. This is because the true wave signals are reciprocal ($\boldsymbol{D}_j$ are symmetric matrices), whereas the additive noise is not. For towed-streamer data, another noise estimation procedure is needed. For example, the measurements at $N_\mathrm{s}' \gg N_\mathrm{s}$ locations, separated by a small distance with respect to the wavelength, can be used to estimate the noise at the N_s receivers used in the ROM construction.

Consider the mass matrices $\boldsymbol{M}(c_o)$ and $\boldsymbol{M}^{\mathrm{N}}(c_o)$ computed by Algorithm 1 from the noiseless background data $\{\boldsymbol{D}_j(c_o)\}_{j=0}^{2N_\mathrm{t}-1}$ and the artificially generated contaminated background data $\{\boldsymbol{D}_j(c_o) + \boldsymbol{E}_j^{\mathrm{N}}\}_{j=0}^{2N_\mathrm{t}-1}$, respectively. Let $\{\sigma_j^o\}_{j=1}^{N_\mathrm{t}N_\mathrm{s}}$ be the singular values of $\boldsymbol{M}(c_o)$, and $\{\sigma_j^{\mathrm{N}}\}_{j=1}^{N_\mathrm{t}N_\mathrm{s}}$ the singular values of $\boldsymbol{M}^{\mathrm{N}}(c_o)$, 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

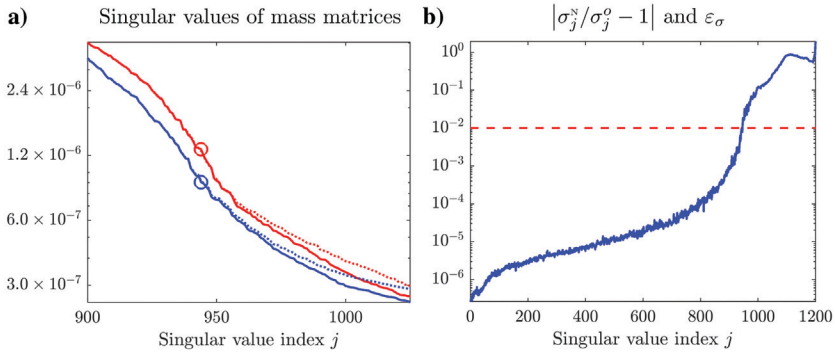

Figure F-1. Regularization threshold illustration: (a) Singular values of mass matrices $\boldsymbol{M}$ (the solid red), $\boldsymbol{M}^{\mathrm{N}}$ (the dotted red), $\boldsymbol{M}(c_o)$ (the solid blue), and $\boldsymbol{M}^{\mathrm{N}}(c_o)$ (the dotted blue). The circles correspond to $j = R^{\mathrm{N}}$. (b) The left side of equation F-2 (the solid blue) and ε_σ (the dashed red).

$$\left| \frac{\sigma_j^{\mathrm{N}}}{\sigma_j^o} - 1 \right| \geq \varepsilon_\sigma. \tag{F-2}$$

Then, we can estimate $r = \lfloor R^{\mathrm{N}}/N_{\mathrm{s}} \rfloor$.

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

In Figure F-1, we illustrate the choice of regularization threshold for the Marmousi model in the setting outlined in the "Numerical results" section ($N_{\mathrm{s}} = 30$, $N_{\mathrm{t}} = 40$, 1% additive noise). Figure F-1a 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 $\boldsymbol{M}^{\mathrm{N}}$. Setting $\varepsilon_\sigma = 10^{-2}$, we obtain $R^{\mathrm{N}} = 944$ from equation F-1, as shown in Figure F-1b. 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 shown in Figure F-1a.

REFERENCES

Borcea, L., V. Druskin, A. V. Mamonov, and M. Zaslavsky, 2014, A model reduction approach to numerical inversion for a parabolic partial differential equation: Inverse Problems, **30**, 125011, doi: 10.1088/0266-5611/30/12/125011.

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, doi: 10.1088/1361-6420/aabb16.

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, doi: 10.1016/j.jcp.2018.12.021.

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, doi: 10.1137/19M1296355.

Borcea, L., J. Garnier, A. Mamonov, and J. Zimmerling, 2021, Reduced order model approach for imaging with waves: Inverse Problems, **38**, 025004, doi: 10.1088/1361-6420/ac41d0.

Borcea, L., J. Garnier, A. Mamonov, and J. Zimmerling, 2022, Waveform inversion with a data driven estimate of the internal wave: arXiv preprint, doi: 10.48550/arXiv.2208.11051.

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, doi: 10.1111/j.1365-246X.2011.04970.x.

Brenner, S., and L. Scott, 2008, The mathematical theory of finite element methods: Springer, Texts in Applied Mathematics 15.

Brossier, R., S. Operto, and J. Virieux, 2010, Which data residual norm for robust elastic frequency-domain full waveform inversion? Geophysics, **75**, no. 3, R37–R46, doi: 10.1190/1.3379323.

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, doi: 10.1073/pnas.1517384113.

Bunks, C., F. Saleck, S. Zaleski, and G. Chavent, 1995, Multiscale seismic waveform inversion: Geophysics, **60**, 1457–1473, doi: 10.1190/1.1443880.

Clayton, R., and R. Stolt, 1981, A Born-WKBJ inversion method for acoustic reflection data: Geophysics, **46**, 1559–1567, doi: 10.1190/1.1441162.

Dines, K., and R. Lytle, 1979, Computerized geophysical tomography: Proceedings of the IEEE, **67**, 1065–1073, doi: 10.1109/PROC.1979.11390.

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, doi: 10.1137/15M1039432.

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, doi: 10.1137/17M1133580.

Engquist, B., and B. Froese, 2014, Optimal transport for seismic full waveform inversion: Communications in Mathematical Sciences, **12**, 979–988, doi: 10.4310/CMS.2014.v12.n5.a7.

Gauthier, O., J. Viricux, and A. Tarantola, 1986, Two-dimensional nonlinear inversion of seismic waveforms: Numerical results: Geophysics, **51**, 1387–1403, doi: 10.1190/1.1442188.

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

Guitton, A., and W. W. Symes, 2003, Robust inversion of seismic data using the Huber norm: Geophysics, **68**, 1310–1319, doi: 10.1190/1.1598124.

Hörmander, L., 2003, The analysis of linear partial differential operators I: Distribution theory and Fourier analysis, 2nd ed.: Springer-Verlag, Classics in Mathematics.

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**, no. 3, R153–R171, doi: 10.1190/geo2016-0301.1.

John, F., 1982, Partial differential equations, 4th ed.: Springer-Verlag.

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, doi: 10.1016/j.cma.2016.03.025.

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, doi: 10.1190/1.1444597.

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

Sava, P., and B. Biondi, 2004, Wave-equation migration velocity analysis. I. Theory: Geophysical Prospecting, **52**, 593–606, doi: 10.1111/j.1365-2478.2004.00447.x.

Stefanov, P., G. Uhlmann, A. Vasy, and H. Zhou, 2019, Travel time tomography: Acta Mathcmatica Sinica, **35**, 1085–1114, doi: 10.1007/s10114-019-8338-0.

Symes, W., 2008, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790, doi: 10.1111/j.1365-2478.2008.00698.x.

Symes, W., and J. J. Carazzone, 1991, Velocity inversion by differential semblance optimization: Geophysics, **56**, 654–663, doi: 10.1190/1.1443082.

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, doi: 10.1111/j.1365-2478.1994.tb00231.x.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266, doi: 10.1190/1.1441754.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, no. 6, WCC1–WCC26, doi: 10.1190/1.3238367.

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**, no. 1, R43–R62, doi: 10.1190/geo2016-0663.1.

Biographies and photographs of the authors are not available.