Waveform inversion via reduced order modeling

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

ABSTRACT

We introduce a novel approach to waveform inversion, based on a data driven reduced order model (ROM) of the wave operator. The presentation is for the acoustic wave equation, but the approach can be extended to elastic or electromagnetic waves. The data are time resolved measurements of the pressure wave gathered by an acquisition system which 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 the step by step computation of the ROM, which depends nonlinearly on the data and yet can be obtained from them in a non-iterative 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 co-located sources and receivers. However, we show that the computation can deal with an approximation of this matrix, obtained from towed-streamer data using interpolation and reciprocity on-the-fly.

While 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 show that the ROM misfit objective function has a better behavior, even for a poor initial guess. We also show by an 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 $N_{\rm s}$ co-located 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 $s^{\rm th}$ source, for $s = 1, \ldots, N_{\rm s}$. It satisfies the wave equation

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

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

for $\boldsymbol{x} \in \Omega$, a simply connected domain, with boundary $\partial\Omega$. This domain can arise from the mathematical truncation of the space, since over the finite duration T of the measurements, the waves are not affected by the medium at distances exceeding $T \max_{\boldsymbol{x}} c(\boldsymbol{x})$. Thus, we can impose any homogeneous boundary conditions at $\partial\Omega$, for example Dirichlet.

The right-hand 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 $(-t_{\rm f}, t_{\rm f})$. 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.

. *University of Michigan, Department of Mathematics, Ann Arbor, MI 48109-1043. †Centre de Mathématiques Appliquées, Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau Cedex, France. ‡University of Houston, Department of Mathematics, Houston, TX 77204-3008. §Uppsala Universitet, Department of Information Technology, Division of Scientific Computing, 75105 Uppsala, Sweden.

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

$$\mathcal{M}^{(r,s)}(t) = p^{(s)}(t, \boldsymbol{x}_r), \quad 1 \le r, s \le N_{\rm s}, \quad t \in (-t_{\rm f}, T]. \tag{3}$$

Note that knowing $\boldsymbol{\mathcal{M}}(t)$ requires co-located 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 travel time tomography (Dines and Lytle, 1979) and its more general version studied in the mathematics community (Stefanov et al., 2019), linearized, a.k.a. Born inversion (Clayton and Stolt, 1981), migration velocity analysis (Symes and Carazzone, 1991; Sava and Biondi, 2004) and full-waveform inversion (Tarantola, 1984; Virieux and Operto, 2009). The first three are based on assumptions such as the velocity changes slowly on the scale of the wavelength (for travel time tomography), or the velocity variations are small (for Born inversion) or there is separation of scales between the smooth components of the velocity and the rough part that gives the reflectivity of the medium (for migration). Full-waveform inversion (FWI) circumvents such assumptions. It is a 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 lead 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) like the differential semblance method (Symes and Carazzone, 1991; Symes and Kern, 1994) and the source-receiver extension method (Huang et al., 2017), introduce in a systematic way additional degrees of freedom in the optimization and then use some objective function to drive the extended model toward a velocity estimate. There are also approaches that use a better alternative than the L^2 norm for measuring the data misfit (Brossier et al., 2010; Bozdağ et al., 2011; Guitton and Symes, 2003). A prominent alternative is the optimal transport (Wasserstein) metric proposed and analyzed for seismic inversion in (Engquist and Froese, 2014; Yang et al., 2018).

We introduce a different approach to velocity estimation, based on a data driven reduced order model (ROM) of the wave operator. The mapping between the measurements 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 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 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

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

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

$$\mathcal{A} = c^{-1}(\boldsymbol{x})\left[-c^2(\boldsymbol{x})\Delta\right] c(\boldsymbol{x}) = -c(\boldsymbol{x})\Delta\big[c(\boldsymbol{x})\cdot\big]. \qquad (5)$$

The approximation that we seek is the ROM wave operator $\partial_t^2 + \boldsymbol{\mathcal{A}}^{\rm ROM}$, where $\boldsymbol{\mathcal{A}}^{\rm ROM}$ 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 $\mathcal{F}$

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

because the composition $\mathcal{R}\circ\mathcal{F}$, which gives $\boldsymbol{\mathcal{A}}^{\rm ROM} = \mathcal{R}\big(\mathcal{F}[c]\big)$, 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

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

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 $\boldsymbol{\mathcal{A}}^{\rm ROM}$ is a Galerkin approximation of $\mathcal{A}$ on the space spanned by the snapshots of the wavefield, at $N_{\rm t}$ discrete and equidistant time instances. Such approximation spaces are common in model order reduction (Brunton and Kutz, 2019; Hesthaven et al., 2016), 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 $\boldsymbol{V}(\boldsymbol{x})$,

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

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

1. The ROM matrix $\boldsymbol{\mathcal{A}}^{\rm ROM}$ can be obtained directly from the measurements $\boldsymbol{\mathcal{M}}(t)$, without knowing the snapshots $\boldsymbol{V}(\boldsymbol{x})$ nor 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 $\boldsymbol{\mathcal{A}}^{\rm ROM}$ in Algorithm 1 and we explain the relationship between the ROM and the data that leads to Algorithm 1 in the next subsection.
2. $\boldsymbol{V}(\boldsymbol{x})$ cannot be computed from the measurements. However, the analysis in (Borcea et al., 2021, Appendix A) and 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 $j^{\rm th}$ time instance are peaked near the maximum depth reached by the wavefield up to this instance.
4. $\boldsymbol{V}(\boldsymbol{x})$ is causal. With the first $k < N_{\rm t}$ snapshots, the definition in equation 8 gives the principal $kN_{\rm s}\times kN_{\rm s}$ submatrix of $\boldsymbol{\mathcal{A}}^{\rm ROM}$, obtained by removing its last $(N_{\rm t}-k)N_{\rm s}$ rows and columns.

Since $\boldsymbol{V}(\boldsymbol{x})$ depends on $c(\boldsymbol{x})$ in a complicated way, we cannot prove the convexity of the ROM misfit objective function $v \mapsto \|\boldsymbol{\mathcal{A}}^{\rm ROM} - \mathcal{R}\big(\mathcal{F}[v]\big)\|_F^2$ 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 travel time 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 $\boldsymbol{\mathcal{A}}^{\rm ROM}$ 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 $\boldsymbol{\Psi}(\boldsymbol{x})$ in equation 7 in terms of $\boldsymbol{V}(\boldsymbol{x})$, the ROM matrix $\boldsymbol{\mathcal{A}}^{\rm ROM}$ 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 above and equation 8 also show that the entries of $\boldsymbol{\mathcal{A}}^{\rm ROM}$ 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 $\boldsymbol{\mathcal{A}}^{\rm ROM}$ behaves similarly to that from the uncomputable $\boldsymbol{\mathcal{A}}^{\Psi}$, which is why we propose using the minimization

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

The minimization problem (9) can be solved with a Gauss-Newton iterative method that is summarized in Algorithm 2. But 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 get

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

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 $(-t_\mathrm{f}, t_\mathrm{f})$ of support of $f(t)$. We assume henceforth that the medium near the co-located sources/receivers is known and homogeneous, with velocity $\bar{c}$. Due to the initial condition in equation 2, we observe that

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

The waves differ at $t\in[0,t_\mathrm{f})$, but since 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 $t\geq -t_\mathrm{f}$, for $s,r=1,\ldots,N_s$, as assumed in equation 3. But even if the measurements are made at $t\geq t_\mathrm{f}$ only, we can still compute $p_e^{(s)}(t,\boldsymbol{x}_r)$ at $t\in[0,t_\mathrm{f})$ 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

$$\begin{aligned} D^{(r,s)}(t) &= p^{(s)}(t,\boldsymbol{x}_r) + p^{(s)}(-t,\boldsymbol{x}_r) \\ &= \mathcal{M}^{(r,s)}(t) + \mathcal{M}^{(r,s)}(-t), \quad 1\leq r,s\leq N_\mathrm{s}. \end{aligned} \tag{12}$$

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 j^{th} snapshot vector field

$$\boldsymbol{u}_j(\boldsymbol{x}) = \left(u^{(1)}(t_j,\boldsymbol{x}),\ldots,u^{(N_\mathrm{s})}(t_j,\boldsymbol{x})\right), \quad \boldsymbol{x}\in\Omega. \tag{13}$$

We are interested only in the first N_t snapshots, and organize them in the $N_\mathrm{t}N_\mathrm{s}$ dimensional row vector field

$$\boldsymbol{U}(\boldsymbol{x}) = \left(\boldsymbol{u}_0(\boldsymbol{x}),\ldots,\boldsymbol{u}_{N_\mathrm{t}-1}(\boldsymbol{x})\right), \quad \boldsymbol{x}\in\Omega. \tag{14}$$

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

$$\boldsymbol{u}_\mathrm{G}(t,\boldsymbol{x}) = \left(u_\mathrm{G}^{(1)}(t,\boldsymbol{x}),\ldots,u_\mathrm{G}^{(N_\mathrm{s})}(t,\boldsymbol{x})\right) = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{g}(t) \tag{15}$$

with time dependent coefficients gathered in the matrices $\boldsymbol{g}(t)\in\mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{s}}$. These coefficients are such that when substituting equation 15 into the homogeneous wave equation, the residual is orthogonal to the approximation space. This gives the following system of second order ordinary differential equations

$$\underbrace{\int_\Omega d\boldsymbol{x}\,\boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x})}_{\boldsymbol{M}}\boldsymbol{g}''(t) + \underbrace{\int_\Omega d\boldsymbol{x}\,\boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{\mathcal{A}}\boldsymbol{U}(\boldsymbol{x})}_{\boldsymbol{S}}\boldsymbol{g}(t) = 0, \tag{16}$$

for $t>0$, with initial condition

$$\boldsymbol{g}(0) = \boldsymbol{e}_0, \quad \boldsymbol{g}'(0) = \boldsymbol{0}. \tag{17}$$

Here $\boldsymbol{e}_0$ is the first $N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{s}$ block of the $N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{t}N_\mathrm{s}$ identity matrix $\boldsymbol{I}_{N_\mathrm{t}N_\mathrm{s}}$. Equation 17 ensures that the Galerkin approximation 15 satisfies the initial conditions

$$\boldsymbol{u}_\mathrm{G}(0,\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{e}_0 = \boldsymbol{u}_0(\boldsymbol{x}), \quad \partial_t\boldsymbol{u}_\mathrm{G}(0,\boldsymbol{x}) = \boldsymbol{0}. \tag{18}$$

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

$$\boldsymbol{M} = \int_\Omega d\boldsymbol{x}\,\boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{U}(\boldsymbol{x}) \in \mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{t}N_\mathrm{s}}, \tag{19}$$

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

$$\boldsymbol{S} = \int_\Omega d\boldsymbol{x}\,\boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{\mathcal{A}}\boldsymbol{U}(\boldsymbol{x}) \in \mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{t}N_\mathrm{s}}, \tag{20}$$

can be calculated directly from $\boldsymbol{D}(t)$ and the second derivative $\boldsymbol{D}''(t)$, evaluated at instances $\{t_j = j\tau\}_{j=0}^{2N_\mathrm{t}-2}$, as explained in the next section (see Appendix C for the estimation of $\boldsymbol{D}''(t)$, using a filtered Fourier transform). Thus, even though we do not know the operator $\boldsymbol{\mathcal{A}}$ 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 16 with the data driven $\boldsymbol{M}$ and $\boldsymbol{S}$, and the initial conditions given in equation 17.

The final step of the ROM construction is to put equation 16 in an algebraic form that describes the evolution of a causal wave $\boldsymbol{u}^{\text{ROM}}(t)\in\mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{s}}$. Each column of this wave corresponds to a source index s, with $1\leq s\leq N_\mathrm{s}$. Initially, the true wave is supported near the sources, which is reflected in the algebraic structure of $\boldsymbol{u}^{\text{ROM}}(0)$, whose only nonzero entries are in the first $N_\mathrm{s}\times N_\mathrm{s}$ block. At later times there is block row fill-in in $\boldsymbol{u}^{\text{ROM}}(t)$, which models wave propagation further away from the sources.

The desired transformation of equation 16 is achieved using the block Cholesky square root (Golub and Van Loan, 2013) of the data driven mass matrix

$$\boldsymbol{M} = \boldsymbol{R}^T\boldsymbol{R}, \tag{21}$$

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

$$\boldsymbol{u}^{\text{ROM}}(t) = \boldsymbol{R}\boldsymbol{g}(t), \tag{22}$$

and we note from equation 17 that at $t = 0$ it satisfies

$$\boldsymbol{u}^{\text{ROM}}(0) = \boldsymbol{R}\boldsymbol{e}_0 = \begin{pmatrix} \boldsymbol{R}_{0,0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix}, \quad \frac{d\boldsymbol{u}^{\text{ROM}}}{dt}(0) = \boldsymbol{0}, \tag{23}$$

where $\boldsymbol{R}_{0,0} \in \mathbb{R}^{N_\mathrm{s}\times N_\mathrm{s}}$. The wave equation in the ROM space is obtained after multiplying equation 16 on the left by $\boldsymbol{R}^{-T} = (\boldsymbol{R}^{-1})^T$,

$$\frac{d^2\boldsymbol{u}^{\text{ROM}}}{dt^2}(t) + \boldsymbol{\mathcal{A}}^{\text{ROM}}\boldsymbol{u}^{\text{ROM}}(t) = 0, \quad t > 0, \tag{24}$$

and the ROM approximation of $\mathcal{A}$ is the $N_\mathrm{t}N_\mathrm{s} \times N_\mathrm{t}N_\mathrm{s}$ matrix

$$\boldsymbol{\mathcal{A}}^{\text{ROM}} = \boldsymbol{R}^{-T}\boldsymbol{S}\boldsymbol{R}^{-1}. \tag{25}$$

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

$$\boldsymbol{U}(\boldsymbol{x}) = \boldsymbol{V}(\boldsymbol{x})\boldsymbol{R}, \tag{26}$$

where $\boldsymbol{V}(\boldsymbol{x})$ is an $N_\mathrm{t}N_\mathrm{s}$ dimensional row vector field, with orthonormal components, i.e., it satisfies

$$\int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{I}_{N_\mathrm{t}N_\mathrm{s}}. \tag{27}$$

This $\boldsymbol{V}(\boldsymbol{x})$ stores the orthonormal basis mentioned earlier in the section. Its causality, in the sense that the j^{th} (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 26 into equation 19, and using equation 27, we observe that $\boldsymbol{R}$ in equation 26 is the same as in equation 21, because

$$\boldsymbol{M} = \boldsymbol{R}^T \int_\Omega d\boldsymbol{x}\, \boldsymbol{V}^T(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x})\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{R}. \tag{28}$$

If we use the Gram-Schmidt equation 26 in equation 25, and recall equation 20 for $\boldsymbol{S}$, we get that $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ satisfies equation 8. Therefore, the data driven $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ defined in equation 25, is in fact the orthogonal projection of the operator $\mathcal{A}$ on the unknown space range$\big(\boldsymbol{U}(\boldsymbol{x})\big)$, 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 26 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 both $\boldsymbol{U}(\boldsymbol{x})$ 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 26 may be viewed as mapping the information in $\boldsymbol{R}$, from the algebraic (ROM) space to the physical space. When we have the wrong kinematics (smooth part of $v(\boldsymbol{x})$), $\boldsymbol{V}(\boldsymbol{x}; v)$ maps the arrivals to incorrect depths. But if the kinematics is only slightly wrong, 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, $\boldsymbol{\mathcal{A}}^{\text{ROM}}$ defined by equation 8 depends on $c(\boldsymbol{x})$ mostly through $\mathcal{A}$ and the objective function of the ROM misfit is locally convex.

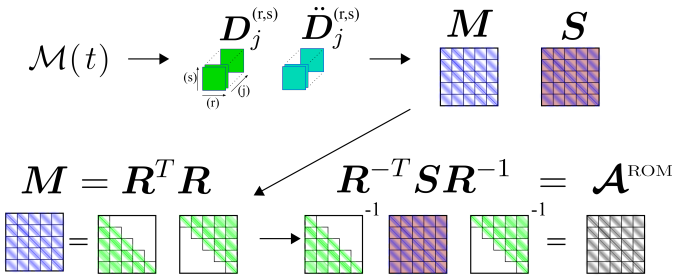

Figure 1: Flow chart for the computation of the ROM from the measurements. There are four steps, each indicated with an arrow. All the matrices are of size $N_\mathrm{t}N_\mathrm{s} \times N_\mathrm{t}N_\mathrm{s}$, with entries organized in $N_\mathrm{s} \times N_\mathrm{s}$ blocks.

Technical details of ROM computation

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

$$\boldsymbol{D}(t) = \Big(\mathcal{M}^{(r,s)}(t) + \mathcal{M}^{(r,s)}(-t)\Big)_{r,s=1}^{N_\mathrm{s}} \tag{29}$$

and their second derivatives $\boldsymbol{D}''(t)$ at instances $t = j\tau$, for $0 \le j \le 2N_\mathrm{t} - 2$. Recall from the previous discussion that $\mathcal{M}^{(r,s)}(-t)$ contributes only at $t = j\tau \in [0, t_\mathrm{f})$ 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 C. Consistent with our previous notation convention, we denote henceforth

$$\boldsymbol{D}_j = \boldsymbol{D}(j\tau), \quad \ddot{\boldsymbol{D}}_j = \boldsymbol{D}''(j\tau). \tag{30}$$

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 38 of the data matrices. These details are not needed to compute $\boldsymbol{\mathcal{A}}^{\text{ROM}}$, 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

$$\frac{[p^{(s)}(t,\boldsymbol{x})+p^{(s)}(-t,\boldsymbol{x})]}{c(\boldsymbol{x})/\bar{c}} = \cos\big(t\sqrt{\mathcal{A}}\big)\widehat{f}\big(\sqrt{\mathcal{A}}\big)\delta_{\boldsymbol{x}_s}(\boldsymbol{x}) = \sum_{j=1}^{\infty}\cos\big(t\sqrt{\lambda_j}\big)\widehat{f}\big(\sqrt{\lambda_j}\big)y_j(\boldsymbol{x}_s)y_j(\boldsymbol{x}), \quad (31)$$

where

$$\widehat{f}(\omega) = \int_{\mathbb{R}} f(t)e^{i\omega t}dt \quad (32)$$

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 $\{\lambda_j\}_{j\geq 1}$ and the eigenfunctions $\{y_j\}_{j\geq 1}$, then $\cos\big(t\sqrt{\mathcal{A}}\big)$ is the operator with eigenvalues $\{\cos\big(t\sqrt{\lambda_j}\big)\}_{j\geq 1}$ and the same eigenfunctions. The operator $\widehat{f}\big(\sqrt{\mathcal{A}}\big)$ is defined similarly. The derivation of equation 31 involves the expansion of the wavefield in the basis $\{y_j(\boldsymbol{x})\}_{j\geq 1}$ of eigenfunctions of $\mathcal{A}$ and manipulations of series.

Next, we need the technical assumption that $\widehat{f} \geq 0$. 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

$$f(t) = \varphi(t) \star_t \varphi(-t). \quad (33)$$

Such $f(t)$ is obviously an even function, with Fourier transform $\widehat{f}(\omega) = |\widehat{\varphi}(\omega)|^2 \geq 0$, 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 hand side in equation 31 as

$$\cos\big(t\sqrt{\mathcal{A}}\big)\widehat{f}\big(\sqrt{\mathcal{A}}\big)\delta_{\boldsymbol{x}_s}(\boldsymbol{x}) = \widehat{f}^{\frac{1}{2}}\big(\sqrt{\mathcal{A}}\big)u^{(s)}(t,\boldsymbol{x}), \quad (34)$$

where

$$u^{(s)}(t,\boldsymbol{x}) = \cos\big(t\sqrt{\mathcal{A}}\big)u_0^{(s)}(\boldsymbol{x}), \quad (35)$$

is our new wave, with initial state

$$u_0^{(s)}(\boldsymbol{x}) = \widehat{f}^{\frac{1}{2}}\big(\sqrt{\mathcal{A}}\big)\delta_{\boldsymbol{x}_s}(\boldsymbol{x}). \quad (36)$$

Note that $u^{(s)}(t,\boldsymbol{x})$ is just like the wave written in equation 31. The only difference is that it corresponds to a different pulse, with Fourier transform $\widehat{f}^{\frac{1}{2}}$ instead of $\widehat{f}$.

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

$$\cos((t+\Delta t)\alpha) = 2\cos(\Delta t\alpha)\cos(t\alpha) - \cos((t-\Delta t)\alpha),$$

for $\alpha = \sqrt{\lambda_j}$, with $j \geq 1$, to evolve the wave defined in equation 35 over any interval Δt,

$$u^{(s)}(t+\Delta t,\boldsymbol{x}) = 2\cos\big(\Delta t\sqrt{\mathcal{A}}\big)u^{(s)}(t,\boldsymbol{x}) - u^{(s)}(t-\Delta t,\boldsymbol{x}). \quad (37)$$

The second consequence is that the entries of $\boldsymbol{D}(t)$, defined in equation 12, admit a useful symmetric inner product expression

$$\begin{aligned} D^{(r,s)}(t) &= p^{(s)}(t,\boldsymbol{x}_r) + p^{(s)}(-t,\boldsymbol{x}_r) \\ &= \int_\Omega d\boldsymbol{x}\, \delta_{\boldsymbol{x}_r}(\boldsymbol{x})\widehat{f}^{\frac{1}{2}}\big(\sqrt{\mathcal{A}}\big)u^{(s)}(t,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\, \big[\widehat{f}^{\frac{1}{2}}\big(\sqrt{\mathcal{A}}\big)\delta_{\boldsymbol{x}_r}(\boldsymbol{x})\big]u^{(s)}(t,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})u^{(s)}(t,\boldsymbol{x}) \\ &= \int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})\cos\big(t\sqrt{\mathcal{A}}\big)u_0^{(s)}(\boldsymbol{x}), \end{aligned} \quad (38)$$

for $1 \leq r,s \leq N_{\rm s}$. The second equality in this equation is from equation 31 and the assumption $c(\boldsymbol{x}_r) = \bar{c}$, the third equality is because $\mathcal{A}$ and therefore $\widehat{f}^{\frac{1}{2}}\big(\sqrt{\mathcal{A}}\big)$ are self-adjoint operators that commute and the last equalities follow from equations 35 and 36. We also have

$$\begin{aligned} \frac{d^2D^{(r,s)}(t)}{dt^2} &= \int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})\partial_t^2 u^{(s)}(t,\boldsymbol{x}) \\ &= -\int_\Omega d\boldsymbol{x}\, u_0^{(r)}(\boldsymbol{x})\mathcal{A}u^{(s)}(t,\boldsymbol{x}), \quad 1 \leq r,s \leq N_{\rm s}. \end{aligned} \quad (39)$$

Now we can describe how we use equations 37–39 to complete the second step in the flow chart of Figure 1. With the notation

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

for the integral of the outer product of any two functions $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\psi}(\boldsymbol{x})$ with values in $\mathbb{R}^{1\times N_{\rm s}}$, and from the definition in equation 19, we compute the $N_{\rm s}\times N_{\rm s}$ blocks of the mass matrix as

$$\begin{aligned} \boldsymbol{M}_{i,j} &= \langle \boldsymbol{u}_i, \boldsymbol{u}_j\rangle = \langle \cos\big(i\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0, \cos\big(j\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0\rangle \\ &= \langle \boldsymbol{u}_0, \cos\big(i\tau\sqrt{\mathcal{A}}\big)\cos\big(j\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0\rangle \\ &= \frac{1}{2}\langle \boldsymbol{u}_0, \big[\cos\big((i+j)\tau\sqrt{\mathcal{A}}\big) + \cos\big(|i-j|\tau\sqrt{\mathcal{A}}\big)\big]\boldsymbol{u}_0\rangle \\ &= \frac{1}{2}\left(\boldsymbol{D}_{i+j} + \boldsymbol{D}_{|i-j|}\right), \quad 0 \leq i,j \leq N_{\rm t}-1. \end{aligned} \quad (40)$$

The second line in this equation is because $\mathcal{A}$ and therefore $\cos\big(i\tau\sqrt{\mathcal{A}}\big)$ are self-adjoint operators that commute, the third line is due to equation 37, evaluated at $t = i\tau$ and $\Delta t = j\tau$, and the last line is by equation 38. The blocks of the stiffness matrix defined in equation 20 are

$$\begin{aligned} \boldsymbol{S}_{i,j} &= \langle \boldsymbol{u}_i, \mathcal{A}\boldsymbol{u}_j\rangle = \langle \cos\big(i\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0, \mathcal{A}\cos\big(j\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0\rangle \\ &= \langle \boldsymbol{u}_0, \mathcal{A}\cos\big(i\tau\sqrt{\mathcal{A}}\big)\cos\big(j\tau\sqrt{\mathcal{A}}\big)\boldsymbol{u}_0\rangle \\ &= \frac{1}{2}\langle \boldsymbol{u}_0, \mathcal{A}\boldsymbol{u}_{i+j} + \mathcal{A}\boldsymbol{u}_{|i-j|}\rangle \\ &= -\frac{1}{2}\left(\ddot{\boldsymbol{D}}_{i+j} + \ddot{\boldsymbol{D}}_{|i-j|}\right), \quad 0 \leq i,j \leq N_{\rm t}-1, \end{aligned} \quad (41)$$

where we used again the self-adjointness of $\mathcal{A}$, and equation 37 evaluated at $t = i\tau$ and $\Delta t = j\tau$. The last equality

is by equation 39. The block structure of the matrices $\boldsymbol{M}$ and $\boldsymbol{S}$ is sketched in Figure 1 for the case $N_\mathrm{t} = 5$.

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.

Algorithm 1 (Data-driven ROM operator)

Input: *The matrix $\boldsymbol{\mathcal{M}}(t)$ of measurements given by 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 C).

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

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

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

ROM based velocity estimation

We estimate $c(\boldsymbol{x})$ by minimizing the misfit of the ROM, as in equation 9. The computation of the term $\mathcal{R}\big(\mathcal{F}[v]\big)$ 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

$$\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v) = \mathcal{R}\big(\mathcal{F}[v]\big). \tag{42}$$

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

$$v(\boldsymbol{x};\boldsymbol{\eta}) = c_o(\boldsymbol{x}) + \sum_{l=1}^{N} \eta_l\phi_l(\boldsymbol{x}), \tag{43}$$

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 43.

The causality of the ROM (Appendix A) 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 $k \le N_\mathrm{t}$. To do so, we replace $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v)$ and $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ in the objective function by the upper left $kN_\mathrm{s} \times kN_\mathrm{s}$ blocks of these matrices, denoted by $\big[\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v)\big]_k$ and $\big[\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}\big]_k$, respectively.

Since $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ and thus $\big[\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}\big]_k$ are symmetric matrices, it is enough to consider their block upper triangular part in the optimization. As shown in Appendix B, the entries of $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ 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

$$\mathrm{Rest}_{d,k} : \mathbb{R}^{kN_\mathrm{s}\times kN_\mathrm{s}} \mapsto \mathbb{R}^{dN_\mathrm{s}(kN_\mathrm{s}-(dN_\mathrm{s}-1)/2)} \tag{44}$$

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

$$\sum_{j=0}^{dN_\mathrm{s}-1} (kN_\mathrm{s} - j) = dN_\mathrm{s}[kN_\mathrm{s} - (dN_\mathrm{s} - 1)/2]. \tag{45}$$

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

$$\mathcal{O}_{d,k}(v) = \big\|\mathrm{Rest}_{d,k}\big(\,[\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v) - \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}]_k\,\big)\big\|_2^2, \tag{46}$$

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

Algorithm 2 (ROM based velocity estimation)

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

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 l^{th} layer is $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k_l-2}$.

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

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

$$\mathcal{L}_i(\boldsymbol{\eta}) = \mathcal{O}_{d,k_l}\big(v(\cdot;\boldsymbol{\eta})\big) + \mathcal{L}_i^{\mathrm{reg}}(\boldsymbol{\eta}), \tag{47}$$

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 $c^{\mathrm{est}}(\boldsymbol{x}) = v(\boldsymbol{x};\boldsymbol{\eta}^{(\ell n_\mathrm{iter})})$.*

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

Computational cost

Since our Algorithm 2 for ROM based velocity estimation uses a Gauss-Newton iteration to minimize the objective function in equation 47, we compare its cost to that of the Gauss-Newton method for minimizing the FWI objective function in equation 4. The same parametrization of the search velocity is assumed for both approaches.

The numerical examples considered below are for two-dimensional media $\Omega \subset \mathbb{R}^2$ with a relatively modest number N_s of co-located 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 $N_{\rm s}$ sources. The ROM based approach requires, in addition, the computation of $\boldsymbol{\mathcal{A}}^{\rm ROM}$ 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 1–2) 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 $\tau_{\rm f}$, and a five point finite difference discretization of the Laplacian on a uniform mesh with $N_{\rm f}$ points. To write down the order of $N_{\rm f}$, 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 $\bar{c}\tau_{\rm f}$. The number of mesh points is therefore

$$N_{\rm f} = \frac{\text{area}(\Omega)}{h^2} \gg N_{\rm s} N_{\rm t},$$

where the inequality is because the co-located 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 $\tau_{\rm f}$. Each time step requires multiplying an $N_{\rm f} \times N_{\rm f}$ sparse matrix with a vector in $\mathbb{R}^{N_{\rm f}}$, at an $O(N_{\rm f})$ cost. Thus, the cost of solving the forward problem, for the $N_{\rm s}$ sources and up to time T, is

$$\text{cost}(\mathcal{F}) = O(N_{\rm s} n_{\rm f} N_{\rm f}),$$

where $n_{\rm f} = T/\tau_{\rm f} \gg N_{\rm t}$. Recall that $\mathcal{F}$ denotes the forward map.

The computational cost of running Algorithm 1 lies mainly in the block Cholesky factorization (see equation 21) and the operator ROM computation from equation 25, where $\boldsymbol{R}^{-1}$ can be calculated by block-wise backward substitution. Therefore, the cost of computing $\boldsymbol{\mathcal{A}}^{\rm ROM}$ is estimated at

$$\text{cost}(\boldsymbol{\mathcal{A}}^{\rm ROM}) = O({N_{\rm s}}^3 {N_{\rm t}}^3),$$

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 below, after equation 48. The bulk of the computational cost of derivatives of $\boldsymbol{\mathcal{A}}^{\rm ROM}$ is in the differentiation of the block Cholesky factors $\boldsymbol{R}$. This cost is essentially the same as that of the block Cholesky factorization itself, since 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 three-dimensional media $\Omega \subset \mathbb{R}^3$ and settings with large $N_{\rm s}$, 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 47. While small-scale examples allow for direct computation of the update direction using, e.g., equation D-4, large-scale settings call for iterative approaches like the Conjugate Gradient method. Note, however, that in such settings the computational cost difference between the conventional FWI and ROM based velocity estimation virtually disappears, since the sizes of the Jacobians of both methods can be made essentially identical by an appropriate choice of parameter d in equation 46.

NUMERICAL ILLUSTRATION

In this section we give two numerical illustrations of the benefits of the velocity estimation with the ROM operator vs. FWI. We assume, as in the theory section above, knowledge of the noiseless array response matrix $\boldsymbol{\mathcal{M}}(t)$. Noisy measurements and the approximation of $\boldsymbol{\mathcal{M}}(t)$ 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 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 the numerical results are for the source pulse

$$f(t) = \cos(\omega_o t) \exp\Big[-\frac{(2\pi B)^2 t^2}{2}\Big], \tag{48}$$

with central frequency $\omega_o/(2\pi) = 6$ Hz and bandwidth $B = 4$ Hz. See Appendix C for details on the numerically simulated data. 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 $N_{\rm s}$ 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 $N_{\rm s}$, the size of the rectangular domain Ω, the data sampling interval τ and the number $N_{\rm t}$ of snapshots that define the approximation space.

Topography of the objective function

Consider the velocity model displayed 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 $c_{\rm t} = 1500$ m/s, while the bottom region has the faster velocity $c_{\rm b} = 3000$ 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 43. 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 $c_{\rm b}/c_{\rm t}$ in the interval $[1, 3]$.

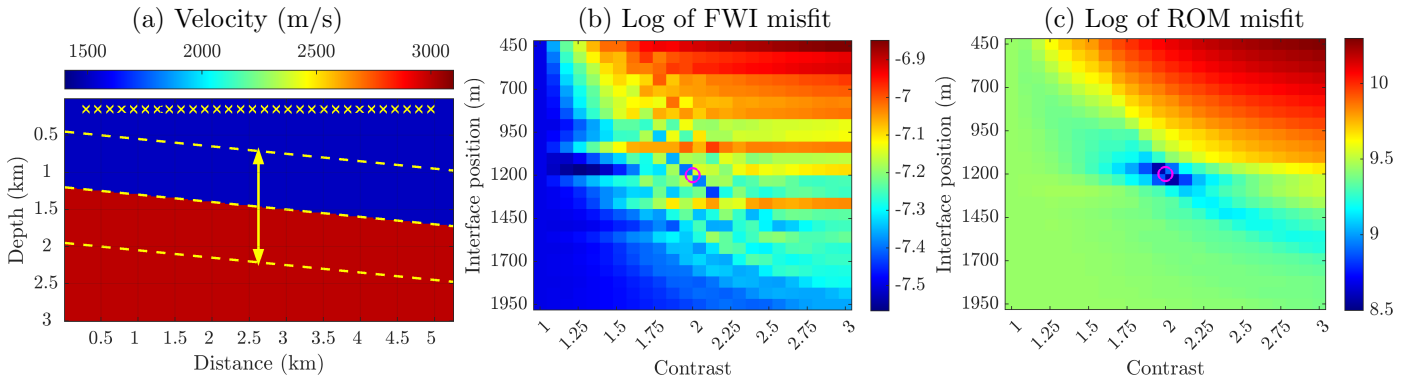

Figure 2: Objective functions topography study: (a) Velocity model used in objective topography study. The middle dashed line shows the actual interface location, while the top and bottom dashed lines show the extent of the interface location parameter sweep. All $N_\mathrm{s} = 30$ co-located sources/receivers are shown as yellow $\times$. Velocity colorbar is in m/s; (b)–(c) Decimal logarithms of the objective functions 49–50, vs. the interface position and velocity contrast. The actual position and contrast parameters are indicated by ◯. These true values are not included in the search space.

The actual contrast is two. The angle of the interface is kept constant and equal to the actual angle.

In Figures 2b–2c we display the decimal logarithms of two objective functions, calculated for $N_\mathrm{s} = 30$ co-located sources/receivers and $N_\mathrm{t} = 39$ time samples at interval $\tau = 0.0435$ s. The first objective function is for the FWI approach,

$$\mathcal{O}^{\mathrm{FWI}}(v) = \sum_{k=0}^{2N_\mathrm{t}-1} \left\| \mathrm{Triu}\big(\boldsymbol{D}_k(v) - \boldsymbol{D}_k\big) \right\|_2^2, \qquad (49)$$

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

$$\mathcal{O}^{\mathrm{ROM}}(v) = \left\| \mathrm{Triu}\big(\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v) - \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}\big) \right\|_2^2. \qquad (50)$$

This corresponds to the particular case $d = k = N_\mathrm{t}$ of the objective function in equation 46.

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 by 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 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 radius of 600 m, centered at point (1 km, 1 km) in the domain $\Omega = [0, 2\ \mathrm{km}] \times [0, 2.5\ \mathrm{km}]$.

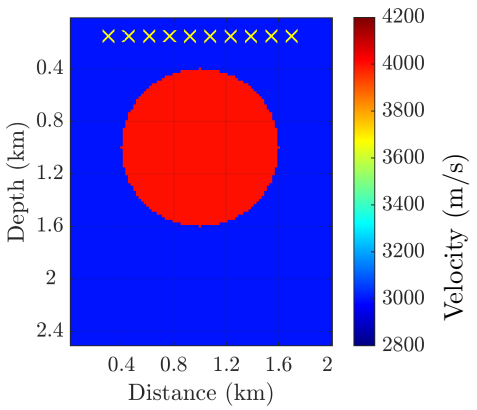

Figure 3: Camembert velocity model. All $N_\mathrm{s} = 10$ co-located sources/receivers are shown as yellow $\times$. Velocity colorbar is in m/s.

The setup is illustrated 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, $N_\mathrm{s} = 10$ and $N_\mathrm{t} = 16$.

The search space $\mathcal{C}$ has dimension $N = 20 \times 20 = 400$, and the velocity is parametrized as in equation 43, with the constant initial guess $c_o(\boldsymbol{x}) = \bar{c} = 3000$ m/s and the Gaussian basis functions

$$\phi_l(\boldsymbol{x}) = \frac{1}{2\pi\sigma_\phi\sigma_\phi^\perp} \exp\left[-\frac{(x^\perp - x_l^\perp)^2}{2(\sigma_\phi^\perp)^2} - \frac{(x - x_l)^2}{2\sigma_\phi^2} \right], \qquad (51)$$

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 over-parametrized

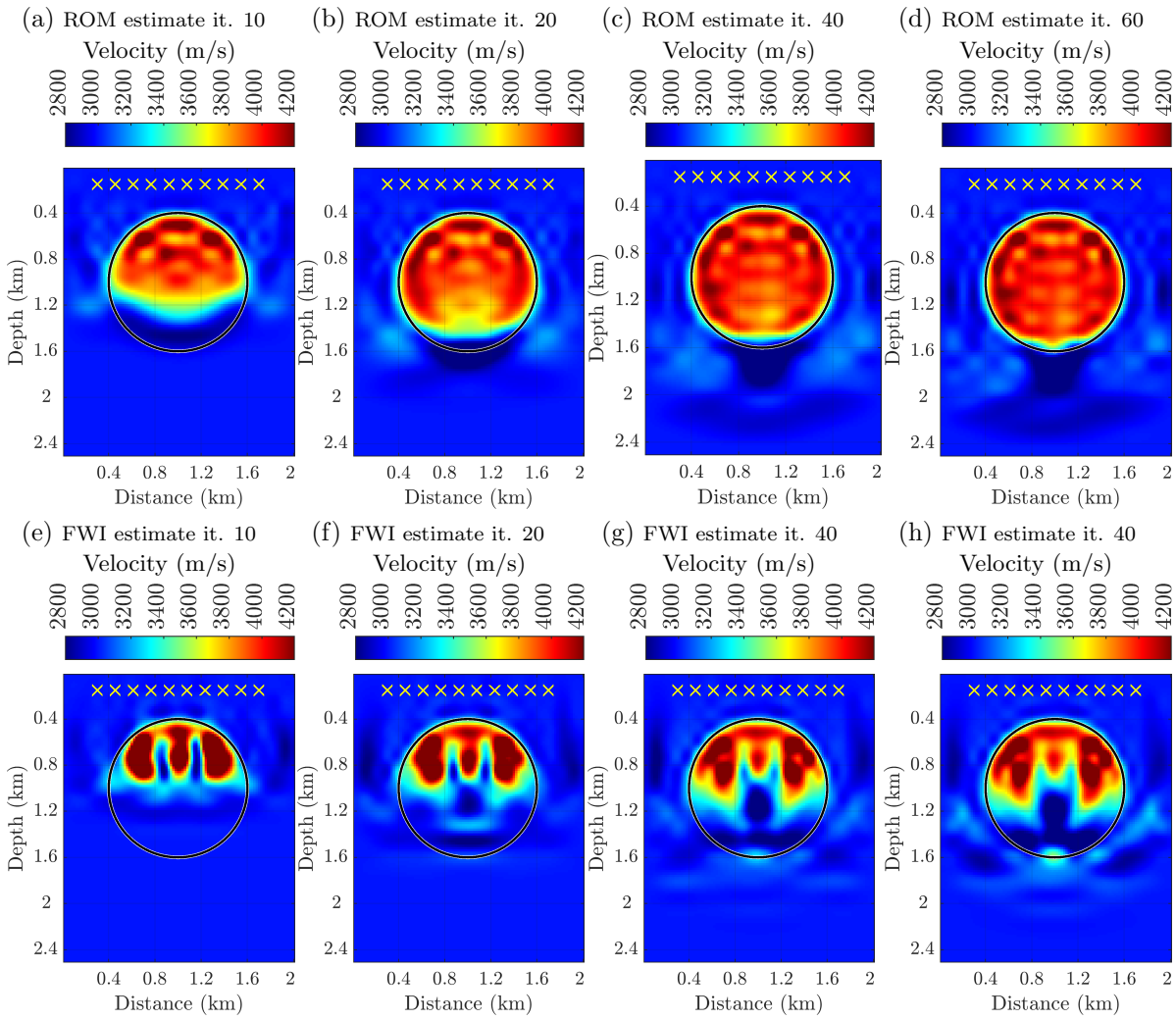

Figure 4: Estimated velocity after $10-60$ Gauss-Newton iterations: (a)–(d) ROM based velocity estimates; (e)–(h) FWI velocity estimates. The true inclusion boundary is shown as a black circle. All $N_{\rm s}=10$ co-located sources/receivers are shown as yellow $\times$. Velocity colorbars are in m/s, all plots share the same color scale.

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 $n_{\rm iter}=4$, and with the restriction parameter $d=N_{\rm t}$. The plots in Figure 4e–4h are the velocity estimates obtained with the FWI approach, which minimizes the objective function

$$\mathcal{L}_i^{\rm FWI}(\boldsymbol{\eta})=\mathcal{O}^{\rm FWI}\big(v(\cdot;\boldsymbol{\eta})\big)+\mu_i^{\rm FWI}\|\boldsymbol{\eta}\|_2^2, \tag{52}$$

with the same time windowing of the data as in the ROM based estimation. The Tikhonov regularization parameter $\mu_i^{\rm FWI}$ 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 $60^{\rm th}$ step, the circular inclusion is reconstructed well. The FWI approach does not improve much after the $10^{\rm th}$ step, indicating that the optimization is stuck in a local minimum. While 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 simply 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 ba-

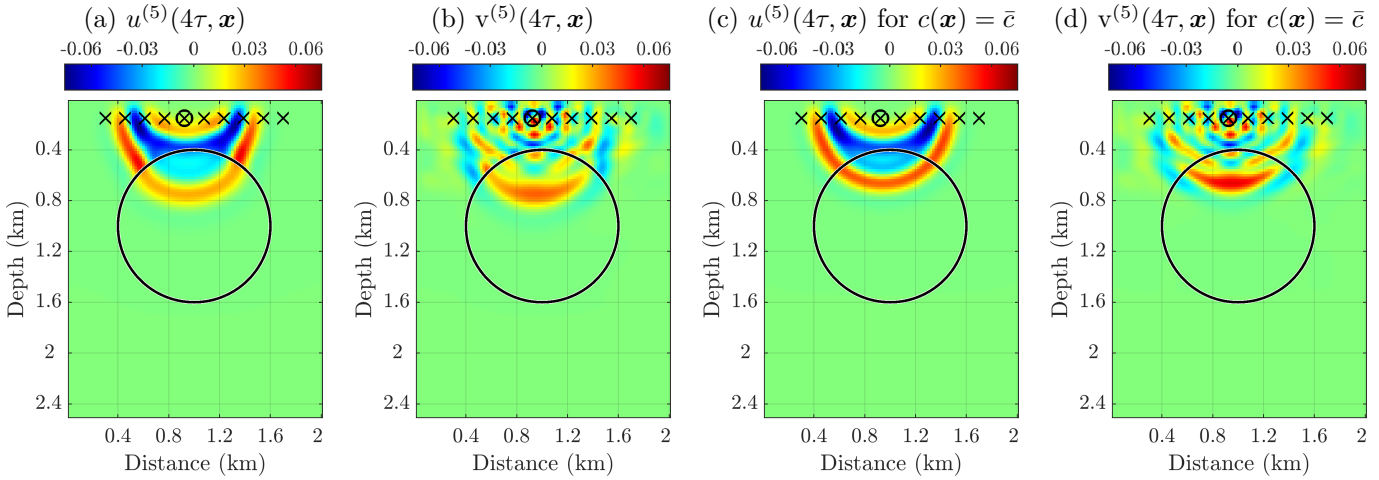

Figure 5: Wavefield snapshots and orthonormal basis components at time instance $t = 4\tau$, corresponding to the center left source, indexed by $s = 5$, shown as a black ◯: (a)–(b) plots for the true velocity $c(\boldsymbol{x})$ displayed in Figure 3; (c)–(d) plots for the reference medium, with $c(\boldsymbol{x})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

sis stored in $\boldsymbol{V}(\boldsymbol{x})$, called $\mathrm{v}^{(5)}(4\tau, \boldsymbol{x})$, are shown in Figures 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 are very similar. They both have a localized peak near the deepest point reached by the wave at 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 above, 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 uncertainty of the data, which affects the computation of $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$. 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 $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ 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

$$\int_{\mathbb{R}} dt\, e^{i\omega t}[\boldsymbol{\mathcal{M}}(t) - \mathcal{F}[\bar{c}](t)]. \tag{53}$$

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

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 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 depth 266 m. The domain is $\Omega = [0, 5.25\ \mathrm{km}] \times [0, 3\ \mathrm{km}]$. The data sampling for the ROM construction is $\tau = 0.0435$ s and the number of snapshots that span the approximation space is $N_\mathrm{t} = 40$. The co-located sources/receivers are located underwater at depth 150 m and they emit the same pulse given in equation 48. We present results in two settings. First, when working with noisy data, we employ an array

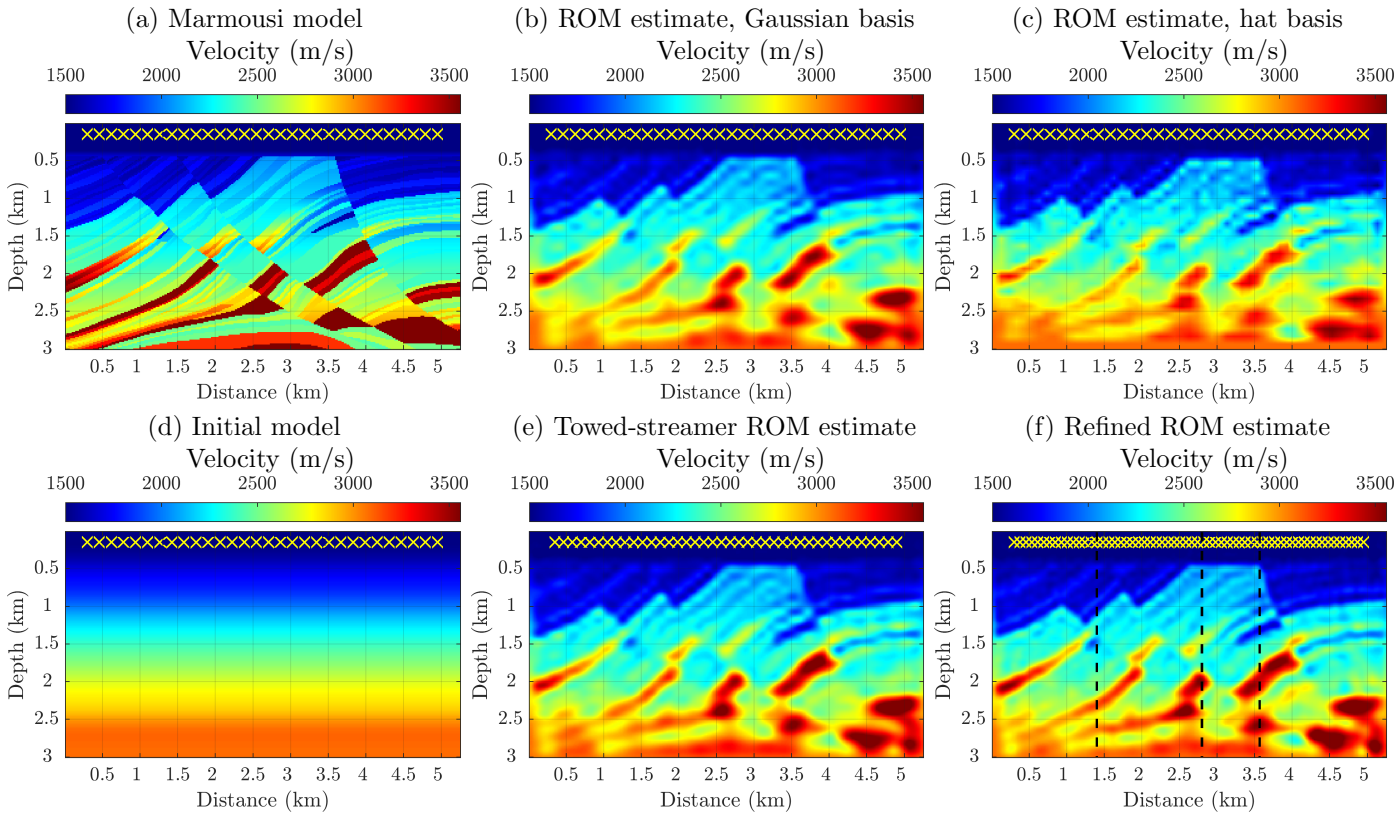

Figure 6: ROM based velocity estimates for Marmousi model with noisy data and towed-streamer measurements: (a) The section of the Marmousi model; (b) Velocity estimate from noisy data with Gaussian basis functions parametrization of v; (c) Velocity estimate from noisy data with hat basis functions parametrization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

of $N_{\rm s} = 30$ co-located sources/receivers, separated by the distance 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 $N_{\rm s} = 40$ sources/receivers separated by the distance 116.66 m.

In Figures 7a–7c we show the ROM based inversion results obtained from data contaminated with 1% additive noise described in Appendix C. We used $\ell = 6$ layers in Algorithm 2, with $n_{\rm iter} = 3$ 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 $r = N_{\rm t} - 9 = 31$. The velocity is parametrized as in equation 43, 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 51, 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 $\Omega_{\rm im} = [103\text{ m}, 5147\text{ m}] \times [97\text{ m}, 2903\text{ m}]$ contained in Ω. Figure 7d–7f shows the FWI results computed for noiseless data. We use the same parametrization 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 $12^{\rm th}$ iteration and the result is far from the true model. Effectively, FWI is stuck in a local minimum.

In Figures 6b–6c we compare the ROM based estimates obtained with two different choices of the basis functions in the parametrization from expression 43 of the search velocity. The Gaussian ones given in equation 51 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 sen-

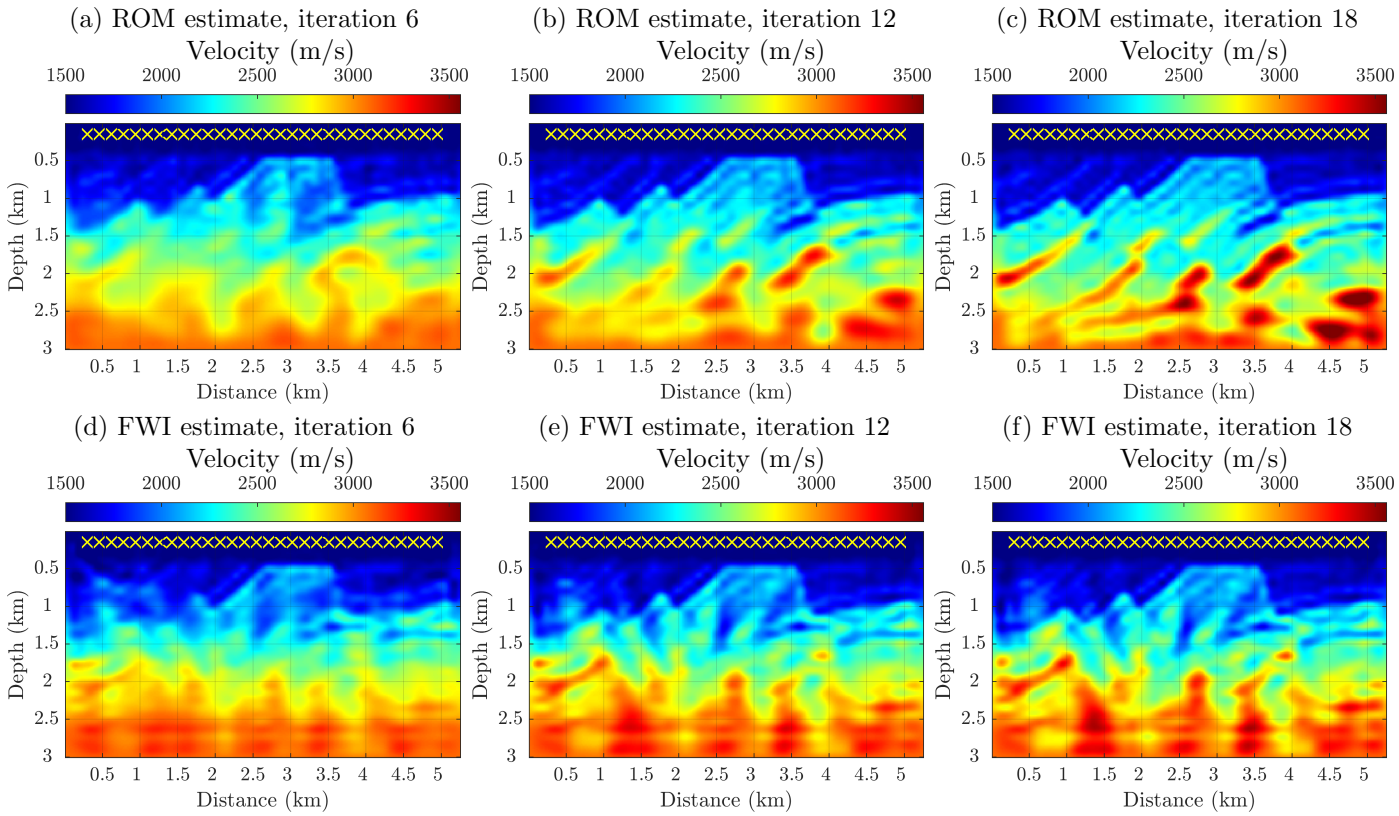

Figure 7: Velocity estimates for the Marmousi model with noisy data after 6, 12 and 18 Gauss-Newton iterations: (a)–(c) ROM based approach; (d)–(f) FWI approach. The $N_s = 30$ co-located sources/receivers are shown as yellow $\times$. Velocity colorbars are in m/s and all plots share the same color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

sitive to the parametrization 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 co-located 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 above, except that the parametrization of the velocity is with $N = 75 \times 38 = 2850$ Gaussian functions with $\sigma_\phi^\perp = 40.2$ m, and $\sigma_\phi = 44.8$ m. We use the estimate from Figure 6b as an initial guess. Since 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 1.4 km, 2.8 km 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 above.

We end the section with the velocity estimate obtained with the array response matrix estimated from towed-streamer type measurements, which is displayed 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 reduced order model (ROM) of the wave operator. The ROM is computed from the data gathered by an array of co-located sources and receivers. Such data can be approximated in geophysics applications from towed-streamer type measurements. No prior information of the medium is used, except for the assumption that the velocity is known in the immediate vicinity of the sensors. While the mapping from the data to the ROM is nonlinear, we can compute it using efficient numerical linear algebra algorithms. We explain that the ROM is an approximation of the wave operator on a space defined by the snapshots of the 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 in order to mitigate additive noise. We demonstrate with numerical simulations that the ROM misfit objective function is better than the nonlinear least-squares data misfit used in full waveform inversion (FWI). In particular, for a

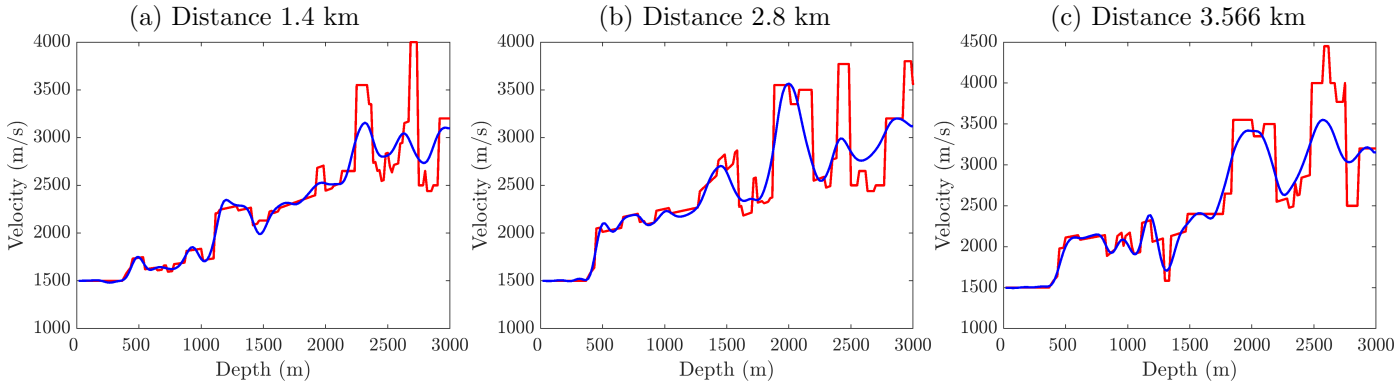

Figure 8: Vertical slices of the Marmousi model velocity (red lines) and its refined ROM estimate (blue lines) at distances shown as dashed lines in Figure 6f.

low-dimensional velocity model where we can plot the objective functions, we obtain that the ROM misfit objective function has convexity properties while the FWI objective function displays multiple local minima. We present velocity estimation results for two well known models where FWI is known to fail in the absence of an excellent initial guess: the "Camembert" model and the Marmousi model.

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 referees 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.

APPENDIX A
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 $\big[\boldsymbol{\mathcal{A}}^{\rm ROM}\big]_k$, is the ROM operator computed by Algorithm 1 from the data subset $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k-2}$, for any $k = 1, \ldots, N_{\rm t}$.

Let us begin by writing $\big[\boldsymbol{\mathcal{A}}^{\rm ROM}\big]_k$ from equation 25

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

where $\boldsymbol{I}_{kN_{\rm s}}$ is the $kN_{\rm s}\times kN_{\rm s}$ identity matrix and $\big[\boldsymbol{S}\big]_k$ and $\big[\boldsymbol{R}\big]_k$ 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 $\big[\boldsymbol{R}\big]_k$.

At step 3, Algorithm 1 computes from $\{\boldsymbol{D}_j, \ddot{\boldsymbol{D}}_j\}_{j=0}^{2k-2}$ the upper left $kN_{\rm s}\times kN_{\rm s}$ block of $\boldsymbol{M}$, denoted by $\big[\boldsymbol{M}\big]_k$, and also $\big[\boldsymbol{S}\big]_k$. The Cholesky factorization in equation 21 and the block upper triangular structure of $\boldsymbol{R}$ give

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

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

APPENDIX B
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 44 to reduce the computational cost of inversion. Let us write

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

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

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

Moreover, the Gram-Schmidt orthogonalization from equation 26 gives

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

and conversely

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

where

$$\boldsymbol{\Gamma} = \boldsymbol{R}^{-1} = \begin{pmatrix} \boldsymbol{\Gamma}_{0,0} & \boldsymbol{\Gamma}_{0,1} & \dots & \boldsymbol{\Gamma}_{0,N_\mathrm{t}-1} \\ \mathbf{0} & \boldsymbol{\Gamma}_{1,1} & \dots & \boldsymbol{\Gamma}_{1,N_\mathrm{t}-1} \\ \vdots & \vdots & \vdots & \boldsymbol{\Gamma}_{N_\mathrm{t}-1,N_\mathrm{t}-1} \end{pmatrix} \tag{B-5}$$

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

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

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

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

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

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

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

and substituting into equation B-6, we obtain

$$\begin{aligned} \boldsymbol{\mathcal{A}}^{\mathrm{ROM}}_{i,j} &= \frac{1}{\tau^2} \sum_{q=0}^{j} \boldsymbol{\Gamma}_{q,j} \Big\{ \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\boldsymbol{x}) \boldsymbol{u}_q(\boldsymbol{x}) \\ &\quad - \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} \int_\Omega d\boldsymbol{x}\, \mathbf{v}_i^T(\boldsymbol{x}) \boldsymbol{u}_{|q-s|}(\boldsymbol{x}) \Big\} \\ &= \frac{1}{\tau^2} \sum_{q=0}^{j} \boldsymbol{\Gamma}_{q,j} \Big\{ \boldsymbol{R}_{i,q} - \sum_{s=-\infty, s\neq 0}^{\infty} \frac{2(-1)^{s+1}}{s^2} \boldsymbol{R}_{i,|q-s|} \Big\}. \end{aligned} \tag{B-8}$$

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

Since $\boldsymbol{\Gamma}_{q,j} = 0$ for $q > j$, and $\boldsymbol{R}_{i,q} = 0$ for $i > q$, the first term on the right-hand side of equation B-8 is zero for $i > j$. But we are interested only in the block upper triangular part of $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ (i.e., $i \leq j$), 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 B-8. Each term in this series adds an s^{th} diagonal, whose entries decay as $1/s^2$. Thus, only the first few block diagonals are large.

APPENDIX C
NUMERICALLY SIMULATED DATA

The data for the numerical experiments are computed with a time-domain wave equation solver for equations 1–2, 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 $\tau_\mathrm{f} = \tau/20$. Using equation 12, we get the finely sampled data $\boldsymbol{D}^\mathrm{f}_k$, for $k = 0, 1, \dots, n_\mathrm{f}$, where $n_\mathrm{f} = 20(2N_\mathrm{t} - 1)$.

The noisy data are computed as follows. Define

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

where b is the desired noise level, e.g., $b = 10^{-2}$ for 1% noise. Then, the contaminated finely sampled data is obtained by adding to $\boldsymbol{D}^\mathrm{f}_k$ a realization of an $N_\mathrm{s} \times N_\mathrm{s}$ random matrix with independent, normally distributed entries with mean zero and standard deviation β for each $k = 1, \dots, n_\mathrm{f}$. Since the data at time zero is computed in the known medium near the co-located sources/receivers, we exclude $k = 0$. To simplify notation, hereafter we denote by $\boldsymbol{D}^\mathrm{f}_k$ both 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 get $\boldsymbol{D}^\mathrm{fe}_j$, $j = -n_\mathrm{f}, \dots, n_\mathrm{f}$, with $\boldsymbol{D}^\mathrm{f}_k = \boldsymbol{D}^\mathrm{fe}_{\pm k}$, $k = 0, 1, \dots, n_\mathrm{f}$. Then, we take the discrete Fourier transform of $\big(\boldsymbol{D}^\mathrm{fe}_j\big)_{j=-n_\mathrm{f}}^{n_\mathrm{f}}$ and differentiate in the Fourier domain after using a sharp cutoff low-pass filter intended to stabilize the calculation. The cutoff frequency is at $\omega_o/(2\pi) + 4B = 22$ Hz. We take the inverse Fourier transform to obtain $\ddot{\boldsymbol{D}}^\mathrm{fe}_j$, at $j = -n_\mathrm{f}, \dots, n_\mathrm{f}$, the finely sampled second derivative data. Finally, we subsample both $\boldsymbol{D}^\mathrm{fe}_j$ and $\ddot{\boldsymbol{D}}^\mathrm{fe}_j$ to get

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

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 43 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

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

be the $N_{\rm s}d(2k-d+1)/2$-dimensional residual vector, whose Euclidean norm squared appears in equation 46. The Jacobian of the objective function evaluated at $\boldsymbol{\eta}=\boldsymbol{\eta}^{(i-1)}$ is the matrix

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

We always choose the parametrization of the velocity from equation 43 so that the Jacobian has more rows than columns. Let $\sigma_1^{(i)}\geq\sigma_2^{(i)}\geq\cdots\geq\sigma_N^{(i)}$ be the singular values of $\boldsymbol{J}^{(i)}$. For a fixed parameter $\gamma\in(0,1)$, with smaller values corresponding to stronger regularization, we set

$$\mu_i=\big(\sigma^{(i)}_{\lfloor\gamma N\rfloor}\big)^2. \tag{D-3}$$

The choice of γ depends on the parametrization in equation 43. Since it is not clear what is the resolution of the inversion, we choose to over-parametrize the velocity, and stabilize the inversion with a small γ, in the range $(0.2,0.4)$. For the results presented in the paper we used $\gamma=0.25$.

The Gauss-Newton update direction for the objective function in equation 47, regularized with $\mathcal{L}_i^{\rm reg}(\boldsymbol{\eta})$ from equation D-1, is

$$\boldsymbol{d}^{(i)}=-\Big(\big(\boldsymbol{J}^{(i)}\big)^T\boldsymbol{J}^{(i)}+\mu_i\boldsymbol{I}_N\Big)^{-1}\big(\boldsymbol{J}^{(i)}\big)^T\boldsymbol{r}^{(i)}, \tag{D-4}$$

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-4 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

$$\alpha^{(i)}=\underset{\alpha\in(0,\alpha_{\max})}{\operatorname{argmin}}\ \mathcal{L}_i\big(\boldsymbol{\eta}^{(i-1)}+\alpha\boldsymbol{d}^{(i)}\big) \tag{D-5}$$

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

$$\boldsymbol{\eta}^{(i)}=\boldsymbol{\eta}^{(i-1)}+\alpha^{(i)}\boldsymbol{d}^{(i)}. \tag{D-6}$$

We use a similar regularization strategy for the FWI objective function in equation 52: If we let $\mathcal{E}^{\rm FWI}(\boldsymbol{\eta})$ be the $N_{\rm t}N_{\rm s}(N_{\rm s}+1)$ dimensional residual vector, with entries

$$\big(\mathcal{E}^{\rm FWI}_j(\boldsymbol{\eta})\big)_{j=kN_{\rm s}(N_{\rm s}+1)/2+1}^{(k+1)N_{\rm s}(N_{\rm s}+1)/2}=\operatorname{Triu}\big(\boldsymbol{D}_k(v)-\boldsymbol{D}_k\big),$$

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

$$\boldsymbol{J}^{{\rm FWI},(i)}=\nabla_{\boldsymbol{\eta}}\mathcal{R}^{\rm FWI}(\boldsymbol{\eta}^{(i-1)})\in\mathbb{R}^{N_{\rm t}N_{\rm s}(N_{\rm s}+1)\times N}, \tag{D-7}$$

where we assume $N\leq N_{\rm t}N_{\rm s}(N_{\rm s}+1)$. Furthermore, for the same fixed parameter γ used in the ROM approach, we set $\mu_i^{\rm FWI}=\big(\sigma^{{\rm FWI},(i)}_{\lfloor\gamma N\rfloor}\big)^2$, where $\big\{\sigma_j^{{\rm FWI},(i)}\big\}_{j=1}^N$ are the singular values of $\boldsymbol{J}^{{\rm FWI},(i)}$, sorted in decreasing order.

APPENDIX E
REGULARIZATION OF THE ROM

Let us denote by $\{\boldsymbol{D}_j^{\rm N}\}_{j=0}^{2N_{\rm t}-1}$ the uncertain data matrices that are either contaminated with noise (see Appendix C) 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 $\boldsymbol{D}_j^{\rm N}$ into $\frac{1}{2}(\boldsymbol{D}_j^{\rm N}+\boldsymbol{D}_j^{{\rm N}\,T})$.

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

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

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

be the eigendecomposition of $\boldsymbol{M}^{\rm N}$, where $\boldsymbol{Z}^{\rm N}$ is the orthogonal matrix of eigenvectors and $\boldsymbol{\Lambda}^{\rm N}=\operatorname{diag}(\lambda_1^{\rm N},\ldots,\lambda_{N_{\rm t}N_{\rm s}}^{\rm N})$ 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). Since we work with $N_{\rm s}\times N_{\rm s}$ blocks, we choose the cut-off at index $rN_{\rm s}$, for integer r satisfying $1\leq r<N_{\rm t}$, and use the first $rN_{\rm s}$ eigenvectors, stored in

$$\boldsymbol{Z}^{{\rm N},r}=(Z_{jl}^{\rm N})_{1\leq j\leq N_{\rm t}N_{\rm s},1\leq l\leq rN_{\rm s}}\in\mathbb{R}^{N_{\rm t}N_{\rm s}\times rN_{\rm s}} \tag{E-2}$$

to define the projected mass matrix

$$\boldsymbol{\Lambda}^{{\rm N},r}=(\boldsymbol{Z}^{{\rm N},r})^T\boldsymbol{M}^{\rm N}\boldsymbol{Z}^{{\rm N},r}=\operatorname{diag}\big(\lambda_1^{\rm N},\ldots,\lambda_{rN_{\rm s}}^{\rm N}\big). \tag{E-3}$$

The resulting $\boldsymbol{\Lambda}^{{\rm N},r}$ is well-conditioned, but it does not have the block Hankel + Toeplitz structure deduced from the causal propagation of the wave (recall equation 40). 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 $\boldsymbol{\mathcal{A}}^{\rm ROM}$ is that its entries decay away from the main diagonal. However, we can get the transformation using another ROM, for the "propagator" operator (Borcea et al., 2018, 2021),

$$\mathcal{P}=\cos\big(\tau\sqrt{\mathcal{A}}\big).$$

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

$$\boldsymbol{u}_{j+1}(\boldsymbol{x})=2\mathcal{P}\boldsymbol{u}_j(\boldsymbol{x})-\boldsymbol{u}_{|j-1|}(\boldsymbol{x}),\quad j\geq 0, \tag{E-4}$$

obtained from equation 37 evaluated at $t=j\tau$ and $\Delta t=\tau$, for $j\geq 0$. The approximation space is the same as in the computation of $\boldsymbol{\mathcal{A}}^{\rm ROM}$, i.e., $\operatorname{range}\big(\boldsymbol{U}(\boldsymbol{x})\big)$. If we let $\widetilde{\boldsymbol{u}}_{\rm G,j}=\boldsymbol{U}(\boldsymbol{x})\widetilde{\boldsymbol{g}}_j$ be the Galerkin approximation at instance

$t = j\tau$, this satisfies the algebraic system of equations

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

for $j \geq 0$. Note how the same data driven mass matrix $\boldsymbol{M}$ appears in this equation. The propagator stiffness matrix $\widetilde{\boldsymbol{S}}$ is also data driven, with $N_\mathrm{s} \times N_\mathrm{s}$ blocks given by

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

where $0 \leq i, j \leq N_\mathrm{t} - 1$. Using the same block Cholesky factorization of $\boldsymbol{M}$ from equation 21 and multiplying the Galerkin equation above by $\boldsymbol{R}^{-1}$, we get the time stepping scheme in the ROM space

$$\widetilde{\boldsymbol{u}}^{\mathrm{ROM}}_{j+1} = 2\boldsymbol{\mathcal{P}}^{\mathrm{ROM}}\widetilde{\boldsymbol{u}}^{\mathrm{ROM}}_j - \widetilde{\boldsymbol{u}}^{\mathrm{ROM}}_{|j-1|}, \tag{E-5}$$

the algebraic analogue of equation E-4, where

$$\widetilde{\boldsymbol{u}}^{\mathrm{ROM}}_j = \boldsymbol{R}\widetilde{\boldsymbol{g}}_j, \quad j \geq 0, \tag{E-6}$$

are the ROM snapshots and

$$\boldsymbol{\mathcal{P}}^{\mathrm{ROM}} = \boldsymbol{R}^{-T}\widetilde{\boldsymbol{S}}\boldsymbol{R}^{-1} = \int_\Omega d\boldsymbol{x}\boldsymbol{V}^T(\boldsymbol{x})\mathcal{P}\boldsymbol{V}(\boldsymbol{x}), \tag{E-7}$$

is the ROM propagator. In the last equality we used the definition of $\widetilde{\boldsymbol{S}}$ and the Gram-Schmidt orthogonalization equation 26.

We refer the interested reader to (Borcea et al., 2020) for a long and detailed analysis of $\boldsymbol{\mathcal{P}}^{\mathrm{ROM}}$. 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 $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$, the matrix $\boldsymbol{\mathcal{P}}^{\mathrm{ROM}}$ is sparse, with block tridiagonal structure. Moreover, $\boldsymbol{\mathcal{P}}^{\mathrm{ROM}}$ is determined by the same mass matrix as $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$. Thus, even though the regularized matrix in equation E-3 is not in the right algebraic form, we can bring it in 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 $\mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times N_\mathrm{t}N_\mathrm{s}}$ and computes an orthogonal basis of $\mathbb{R}^{N_\mathrm{t}N_\mathrm{s}}$ that puts the matrix in block tridiagonal form.

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

$$\widetilde{\boldsymbol{S}}^{\mathrm{N},r} = (\boldsymbol{Z}^{\mathrm{N},r})^T\widetilde{\boldsymbol{S}}^{\mathrm{N}}\boldsymbol{Z}^{\mathrm{N},r} \in \mathbb{R}^{rN_\mathrm{s}\times rN_\mathrm{s}}, \tag{E-8}$$

and we compute

$$\boldsymbol{P}^{\mathrm{N},r} = (\boldsymbol{\Lambda}^{\mathrm{N},r})^{-1/2}\widetilde{\boldsymbol{S}}^{\mathrm{N},r}(\boldsymbol{\Lambda}^{\mathrm{N},r})^{-1/2} \in \mathbb{R}^{rN_\mathrm{s}\times rN_\mathrm{s}}. \tag{E-9}$$

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

$$\boldsymbol{\mathcal{P}}^{\mathrm{ROM},r} = (\boldsymbol{Q}^{\mathrm{N},r})^T\boldsymbol{P}^{\mathrm{N},r}\boldsymbol{Q}^{\mathrm{N},r} \in \mathbb{R}^{rN_\mathrm{s}\times rN_\mathrm{s}} \tag{E-10}$$

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

The matrix $\boldsymbol{\mathcal{P}}^{\mathrm{ROM},r}$ itself is irrelevant for 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

$$(\boldsymbol{Q}^{\mathrm{N},r})^T\boldsymbol{\Lambda}^{\mathrm{N},r}\boldsymbol{Q}^{\mathrm{N},r} = (\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{M}^{\mathrm{N}}\boldsymbol{\Pi}^{\mathrm{N},r} = (\boldsymbol{R}^{\mathrm{N},r})^T\boldsymbol{R}^{\mathrm{N},r}, \tag{E-11}$$

where

$$\boldsymbol{\Pi}^{\mathrm{N},r} = \boldsymbol{Z}^{\mathrm{N},r}\boldsymbol{Q}^{\mathrm{N},r} \in \mathbb{R}^{N_\mathrm{t}N_\mathrm{s}\times rN_\mathrm{s}}. \tag{E-12}$$

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

$$\boldsymbol{\mathcal{A}}^{\mathrm{ROM},r} = (\boldsymbol{R}^{\mathrm{N},r})^{-T}(\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{S}^{\mathrm{N}}\boldsymbol{\Pi}^{\mathrm{N},r}(\boldsymbol{R}^{\mathrm{N},r})^{-1}. \tag{E-13}$$

Equation E-13 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-12: Let $\boldsymbol{M}(v)$ and $\boldsymbol{S}(v)$ be the mass and stiffness matrices calculated as in step 3 of Algorithm 1 from the data computed numerically in the medium with velocity $v(\boldsymbol{x},\boldsymbol{\eta})$. We compute the block Cholesky factorization

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

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

$$\boldsymbol{\mathcal{A}}^{\mathrm{ROM},r}(v) = \boldsymbol{R}^r(v)^{-T}(\boldsymbol{\Pi}^{\mathrm{N},r})^T\boldsymbol{S}(v)\boldsymbol{\Pi}^{\mathrm{N},r}\boldsymbol{R}^r(v)^{-1}. \tag{E-15}$$

The velocity inversion is carried out as in Algorithm 2, with $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}$ and $\boldsymbol{\mathcal{A}}^{\mathrm{ROM}}(v)$ in equation 46 replaced by the regularized $\boldsymbol{\mathcal{A}}^{\mathrm{ROM},r}$ and $\boldsymbol{\mathcal{A}}^{\mathrm{ROM},r}(v)$. Note that the matrix with orthogonal columns $\boldsymbol{\Pi}^{\mathrm{N},r}$ used in equations E-14 and E-15 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-14 is not identical to the one computed with Algorithm 1. Nevertheless, they behave the same with respect to the inversion, as illustrated in Figure F-1, where we plot the logarithm of the objective function

$$\mathcal{O}^{\text{ROM},r}(v) = \left\| \text{Triu}\left(\boldsymbol{\mathcal{A}}^{\text{ROM},r}(v) - \boldsymbol{\mathcal{A}}^{\text{ROM},r}\right) \right\|_2^2 \qquad \text{(E-16)}$$

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

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-15. The idea is that r can be determined from the part of the spectrum of the mass matrix $\boldsymbol{M}^\text{N}$ that is perturbed by the uncertainty. This can be estimated using the mass matrix $\boldsymbol{M}^\text{N}(c_o)$ corresponding to the initial guess velocity $c_o(\boldsymbol{x})$, and perturbed in a similar way.

For noisy sensor array measurements, the matrices

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

can be considered as realizations of the additive noise. This is because the true wave signals are reciprocal ($\boldsymbol{D}_j$ are symmetric matrices), while the additive noise is not. For towed-streamer data other noise estimation procedure is needed. For example, measurements at $N_\text{s}' \gg N_\text{s}$ 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 $\boldsymbol{M}^\text{N}(c_o)$ computed by Algorithm 1 from the noiseless background data $\{\boldsymbol{D}_j(c_o)\}_{j=0}^{2N_\text{t}-1}$ and the artificially generated contaminated background data $\{\boldsymbol{D}_j(c_o) + \boldsymbol{E}_j^\text{N}\}_{j=0}^{2N_\text{t}-1}$, respectively. Let $\{\sigma_j^o\}_{j=1}^{N_\text{t}N_\text{s}}$ be the singular values of $\boldsymbol{M}(c_o)$, and $\{\sigma_j^\text{N}\}_{j=1}^{N_\text{t}N_\text{s}}$ the singular values of $\boldsymbol{M}^\text{N}(c_o)$, sorted in decreasing order. Choose a small ε_σ, the largest relative deviation of singular values past which we consider them contaminated by noise. Let R^N be the smallest among j such that

$$\left|\frac{\sigma_j^\text{N}}{\sigma_j^o} - 1\right| \geq \varepsilon_\sigma. \qquad \text{(F-2)}$$

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

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

In Figure F-2 we illustrate the choice of regularization threshold for the Marmousi model in the setting outlined in the numerical results section ($N_\text{s} = 30$, $N_\text{t} = 40$, 1% additive noise). Figure F-2a shows the singular values σ_j^o and σ_j^N for a range $j = 900, 901, \ldots, 1025$, while also comparing them to the singular values of $\boldsymbol{M}$ and $\boldsymbol{M}^\text{N}$. Setting $\varepsilon_\sigma = 10^{-2}$, we obtain $R^\text{N} = 944$ from equation F-2, as shown in Figure F-2b. 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 $\boldsymbol{M}^\text{N}$ diverge from those of $\boldsymbol{M}$, as observed in Figure F-2a.

REFERENCES

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

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

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

———, 2018, Untangling the nonlinearity in inverse scattering with data-driven reduced order models: Inverse Problems, **34**, 065008.

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

———, 2022, Waveform inversion with a data driven estimate of the internal wave: arXiv preprint 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.

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

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

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

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

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

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

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

Druskin, V., A. Mamonov, A. Thaler, and M. Zaslavsky, 2016, Direct, nonlinear inversion algorithm for hyper-

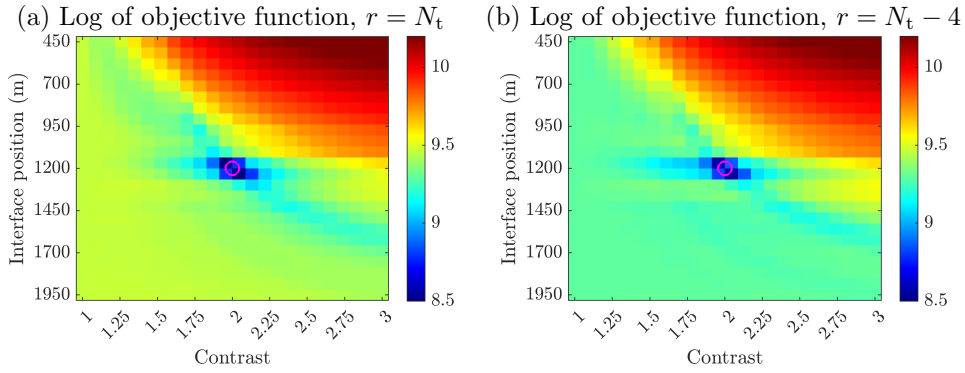

Figure F-1: Decimal logarithm of objective function (equation E-16) vs. the interface position and velocity contrast. The true parameters (shown in Figure 2) are indicated by ◯.

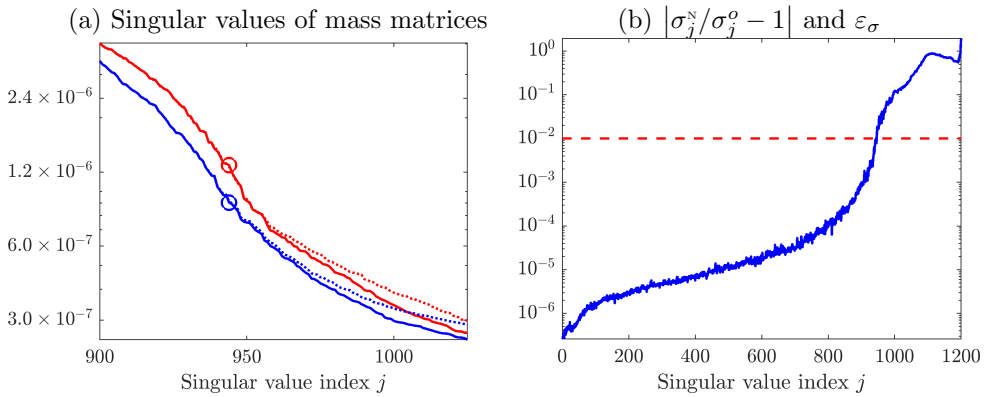

Figure F-2: Regularization threshold illustration: (a) Singular values of mass matrices $\boldsymbol{M}$ (solid red), $\boldsymbol{M}^N$ (dotted red), $\boldsymbol{M}(c_o)$ (solid blue) and $\boldsymbol{M}^N(c_o)$ (dotted blue). The circles correspond to $j = R^N$; (b) Left-hand side of equation F-2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

bolic problems via projection-based model reduction: SIAM Journal on Imaging Sciences, **9**, 684–747.

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

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

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

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

Guitton, A., and W. W. Symes, 2003, Robust inversion of seismic data using the huber norm: Geophysics, **68**, 1310–1319.

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

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

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

John, F., 1982, Partial differential equations, 4 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.

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

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

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

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

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

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

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

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

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

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