Computational Geosciences (2019) 23:829–847
https://doi.org/10.1007/s10596-019-09832-9

ORIGINAL PAPER

Enabling numerically exact local solver for waveform inversion—a low-rank approach

Rajiv Kumar[1] · Bram Willemsen[2] · Felix J. Herrmann[3] · Alison Malcolm[4]

Received: 11 September 2018 / Accepted: 15 May 2019 / Published online: 27 June 2019

Abstract

As in many fields, in seismic imaging, the data in the field is collected over a relatively large medium even though only a part of that medium is truly of interest. This results in significant waste in computation as a typical inversion algorithm requires many solutions of the wave equation throughout the entire domain, even if only a small part of the domain is being updated. One way to mitigate this is to use a numerically exact local wave equation solver to perform waveform inversions in an area of interest, where the idea is to compute accurate solutions of the wave equation within a subdomain of interest. Although such solvers exist, many require the computation of Green's function matrices in the background domain. For large-scale seismic data acquisition, the computation of the Green's function matrices is prohibitively expensive since it involves solving thousands of partial differential equations in the background model. To mitigate this, in this work, we propose to exploit the low-rank structure of the full subsurface Green's function. Using carefully selected 2D stylized models, we first show that the full subsurface Green's function tensor organized as a matrix exhibits the low-rank structure in a transform domain. We then propose a randomized singular value decomposition–based framework to compute the low-rank approximation of the Green's functions, where the cost of wave equation solves depends on the rank of the underlying Green's function matrix instead of the number of grid points at the surface of the background model and on the boundary of the local domain. Next, we validate the proposed framework by performing time-lapse waveform inversion using the 2D Marmousi model. Finally, we demonstrate a rank-minimization–based framework to compute the low-rank factorized form of the Green's function matrices in large-scale 3D seismic data acquisition.

Keywords Low-rank · Randomized SVD · Time-lapse inversion · Local solver · Finite-difference · Wave equation

1 Introduction

Full-waveform inversion (FWI) is a computationally expensive process to predict the properties of a medium of interest, such as the Earth. It is a non-linear process of fitting the observed and simulated data (see [56] for an overview of FWI). To do this, we solve the following optimization problem in the frequency domain

$$\mathbf{m} = \underset{\mathbf{m}}{\operatorname{argmin}} \quad \frac{1}{2}\sum_{j=1}^{N_f}\sum_{i=1}^{N_s} \|\mathbf{P}_r\mathbf{H}_j(\mathbf{m})^{-1}\mathbf{q}_i - \mathbf{d}_i\|_2^2, \tag{1}$$

where $\mathbf{H}_j = (\omega_j^2\mathbf{m} + \nabla^2)$ is a discretization of the Helmholtz operator for constant-density, ω_j represents the temporal frequency for frequency index j, and $\mathbf{m} = \mathbf{v}^{-2}$ is the gridded squared slowness with $\mathbf{v}$ being the unknown spatially varying velocity. The operator $\mathbf{P}_r$ maps the computed wavefield $(\mathbf{H}(\mathbf{m})^{-1}\mathbf{q}_i)$ in the full subsurface domain to the receiver locations, $\mathbf{q}_i$ represents the i-th source, and $\mathbf{d}_i$ is the observed data. Note that seismic data acquisition involves either placing the receivers on the surface/seafloor of the Earth, or towing behind a marine seismic vessel. The acquisition also involves a source (dynamite, airgun, vibroseis), which generates acoustic or elastic vibrations that pass through the complex geological

✉ Rajiv Kumar
rajmittal09@gmail.com

[1] School of Earth and Atmospheric Science, Georgia Institute of Technology, Atlanta, GA, USA

[2] Formerly Earth Resources Laboratory, Massachusetts Institute of Technology; presently ExxonMobil (i.e., ExxonMobil Upstream Research Company), Irving, TX, USA

[3] School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA

[4] Department of Earth Sciences, Memorial University of Newfoundland, St. John's, NL, Canada

structures in the Earth, and return to the surface to be recorded as seismic data [55].

The major computational cost in waveform inversion is solving the Helmholtz equations (or the wave equation in the time domain), which is directly proportional to the number of sources present in an acquisition. We typically solve this optimization problem with some form of Newton iteration. Calculating the necessary gradient involves the computation of forward and adjoint wavefields followed by the cross-correlation of these wavefields for each experiment [45]. Thus, at each iteration, we need to solve two partial differential equations (PDEs) for each source to determine the gradient update. There are several ways to reduce the total computation time such as reducing the number of wave solves via source encoding [16, 31, 34], and improving the Helmholtz solver [32, 44]. Although these techniques can greatly reduce the turn around time to produce good quality inversion results, the experimental demonstration in seismic literature assumes that we are interested in the full subsurface domain to estimate the physical parameters. This is especially not true for time-lapse or updating salt boundaries using FWI, where our area of interest is small (e.g., the reservoir). Note that updating the salt boundary in a velocity model is a labor-intensive [14] imaging method in seismic data processing, which involves months of manual guidance. Hence, various automatic methods for determining the correct salt geometry are proposed in the seismic literature [22, 33, 58]. Moreover, in the time-lapse case [20, 28, 35], we may have more than one monitor dataset, which is acquired over different time intervals. Here, time-lapse data refers to the seismic data acquired at different times over the same area, where the aim is to assess the subsurface changes, such as fluid movement in the reservoir.

Another way to speedup FWI is to solve the wave-equation in a subset of the entire domain. This area of research is divided into two main categories. The first one is known as redatuming [5, 6, 8, 12, 17, 24, 42, 43, 52, 54], where we propagate the seismic data to the region of interest. One of the biggest disadvantages of the redatuming formulation is that it modifies the surface data recordings in the region of interest. To overcome this limitation, in the second approach, we compute the wavefield exactly within the region of interest, which is then used to update the velocity model locally. This approach is known as a numerically exact local solver [37, 39, 50, 60] where the idea is to use the local nature of the FWI problem to speed up the inversion process. Recently, [40, 58] showed the advantages of such a numerically exact local solver for performing FWI in a small area (subdomain) around the region of interest, while still taking all of the data into account. The core idea is to solve both the forward and adjoint wave equation exclusively in the area of interest, giving exactly the same wavefield (on that subdomain) as would be obtained when solving the full-wave equation in the full subsurface domain. There are two main aspects of this type of local solver. First, we need to simulate the Green's function in the full domain of the background model at the surface and on the boundary of the subdomain. We only compute these Green's functions once at the start of the process. After finishing this stage, we no longer need to do full domain simulations anymore during any of the subsequent inversions. Second, we evaluate the forward and adjoint wavefields in the local domain at a significantly reduced cost. Malcolm and Willemsen [36] further illustrate the benefits of the local solver for time-lapse FWI where changes in the velocity around the reservoir are estimated in very few iterations while controlling the computational cost. Note that the underlying assumption of this method is that the model is only changing in the subdomain. When applied to the time-lapse problem, this implies that everything is constant outside the subdomain for all of the datasets.

Although waveform inversion using a local solver is conceptually appealing, it requires the computation of a substantial number of partial differential equations (PDEs) at the beginning of the inversion using the background smooth velocity model for all the frequencies. These solutions are then used to build the Green's function matrices for the local solver, which are required to invert the local solver system of equations. This process is prohibitively expensive for practical 3D seismic surveys, since this requires many costly full domain simulations. The required number of full domain simulations is directly proportional to the number of nodes around the local domain. Therefore, both the computational cost and memory requirements grow drastically when scaling up the local solver from 2D to 3D.

Recently, [2, 23, 25, 53] showed that seismic data generated during 2D and 3D acquisitions exhibit low-rank structure when organized in a transform domain where sources and receivers are sorted appropriately. They then exploit this structure for the compression and/or reconstruction of missing data. Zhang et al. [61] show the advantages of using the low-rank representation of the seismic data in full-waveform inversion. In [21], authors used the low-rank representation of the surface Green's function to reduce the matrix-matrix multiplication cost in the estimation of primaries by sparse inversion framework. Following similar ideas, we propose to exploit the low-rank structure of the full subsurface Green's function to overcome the computational bottleneck of the local solver where we simulate a large number of PDEs in the background velocity model. We present a computationally cost-effective and memory-efficient framework, where we reconstruct the densely sampled Green's function matrices of the local solver with a small percentage of the standard number of simulations. Using the complex 2D Marmousi model [7], we illustrate the efficacy of the proposed approach to perform subdomain time-lapse inversion. We further present a way to exploit the low-rank structure of the Green's function for 3D seismic data acquisition and evaluate it using a synthetic dataset generated on the 3D SEG/EAGE overthrust model.

2 Numerically exact local solver

We begin by explaining the mathematical formulation of using the numerically exact local solver to solve the wave-equation in the subdomain of interest. For simplicity, we explain the numerically exact local solver formulation in 2D. We can follow the same analogy to extend this to 3D. Let Ω represent the interior of the subdomain of interest and Ω^c its exterior. The exterior domain along with the initial guess of the velocity model in the interior of the truncated domain represent the background velocity model $\mathbf{m}_0$. The model perturbation is denoted by $\delta\mathbf{m}$ inside Ω and is zero in Ω^c. Thus, the perturbed model $\mathbf{m} \in \mathbb{R}^{N_x \times N_z}$ in the full domain is defined as $\mathbf{m} = \mathbf{m}_0 + \delta\mathbf{m}$, where N_x and N_z are the number of gridpoints in the x and z directions.

To achieve the objective of solving FWI inside Ω as would be solved in the full subsurface domain, we need a numerically exact expression of the scattered wavefield $\delta\mathbf{u}_s(x)$. Here, we define the scattered wavefield as $\delta\mathbf{u}_s(x) = \mathbf{u}(x) - \mathbf{u}_0(x)$, where $\mathbf{u}(x)$ represents the wavefield in the perturbed model $\mathbf{m}$ and $\mathbf{u}_0(x)$ is the wavefield in the background model $\mathbf{m}_0$. To obtain an accurate wavefield within the truncated domain Ω, we can simply solve Eq. 1 inside Ω. However, to make sure that the predicted solution inside Ω exactly matches the global wavefield solution computed using $\mathbf{m}$, we need to determine the exact conditions on the boundary nodes of the truncated domain.

Fortunately, we can use the representation theorem [15], which states that the scattered field outside a domain can be uniquely and completely determined by the field on the boundary. Masson et al. [38] and Willemsen et al. [58] show that for the local solver, the representation theorem corresponds to relating the scattered field $\delta\mathbf{u}_s(x)$ to the field and its normal derivate on the boundary nodes of the truncated domain. They also show that the estimated scattered field includes all orders of multiple scattering between the interior and exterior of the truncated domain without any loss of accuracy in the local computation. Mathematically, we write the representation theorem for the local solver in the discrete form as

$$\delta\mathbf{u}(y,\omega) = -\sum_{x\in\delta\Omega}\frac{1}{h^2}\Big(\mathbf{u}^{\delta\Omega}(x,\omega)\mathbf{G}_0^{\delta\Omega_{+1}}(x,y,\omega) \\ -\mathbf{u}^{\delta\Omega_{+1}}(x,\omega)\mathbf{G}_0^{\delta\Omega}(x,y,\omega)\Big)\,, \tag{2}$$

where h is the distance between adjacent gridpoints, $\delta\Omega$ is the boundary between Ω and Ω^c, $\delta\Omega_{+1}$ is the gridpoints one layer to the inside of $\delta\Omega$, and $\mathbf{u}^{\delta\Omega}(x,\omega)$ and $\mathbf{u}^{\delta\Omega_{+1}}(x,\omega)$ represent the wavefields on the boundary nodes and on the nodes sitting at $\delta\Omega_{+1}$, respectively. The matrices $\mathbf{G}_0^{\delta\Omega}(x,y,\omega)$ and $\mathbf{G}_0^{\delta\Omega_{+1}}(x,y,\omega)$ represent the background Green's functions between the node y where the scattered field is evaluated and the node x on $\delta\Omega$ and $\delta\Omega_{+1}$, respectively. Appendix 1 describes a process for extracting the Green's function matrices from the full subsurface Green's functions simulated using the sources placed on the boundaries of the truncated domain. For a detailed derivation of Eq. 2, we refer the interested reader to [58]. Using Eqs. 1 and 2, we define a modified system for the wave equation within the local domain,

$$\left(\begin{array}{c|c|cc} -\mathbf{I} & \mathbf{I} & & 0 \\ \hline \mathbf{I} & \mathbf{G}_0^{\delta\Omega_{+1}} & -\mathbf{G}_0^{\delta\Omega} & \\ \hline \mathbf{0} & & \multicolumn{2}{l}{-\omega_j^2\mathbf{m}^{\Omega} - \nabla^2} \end{array}\right)\begin{pmatrix}\delta\mathbf{u}^{\delta\Omega}\\ \mathbf{u}^{\delta\Omega}\\ \mathbf{u}^{\delta\Omega_{+1}}\\ \mathbf{u}^{\delta\Omega_{+2}}\\ \vdots\end{pmatrix} = \begin{pmatrix}\mathbf{u_0}^{\delta\Omega}\\ \hline \mathbf{0}\\ \hline \mathbf{0}\\ \mathbf{0}\\ \vdots\end{pmatrix}, \tag{3}$$

which we use to estimate the exact wavefields in the truncated domain of the perturbed model $\mathbf{m}$. Here, $\mathbf{u_0}^{\delta\Omega}$ is the forward (adjoint) wavefields, which we compute by convolving the Green's function with the source wavelet (data residual), where the Green's function is recorded at the receivers placed on the boundary of the truncated domain for the sources placed at the surface of the full domain. We want to emphasize again that the model $\mathbf{m}^{\Omega} = \delta\mathbf{m}^{\Omega} + \mathbf{m}_0^{\Omega}$ in Eq. 3 where $\delta\mathbf{m}^{\Omega}$ represents the perturbation of the true model from the background model ($\mathbf{m}_0^{\Omega}$) restricted to the local domain $\delta\Omega$. Here, the size of $\mathbf{m}^{\Omega}$, $\delta\mathbf{m}^{\Omega}$, and $\mathbf{m}_0^{\Omega}$ is $N_{x_{sub}} \times N_{z_{sub}}$. To form (3), we follow the same node numbering scheme inside the truncated domain as proposed in [58] on which we give more details in Appendix 1. We also provide details in Appendix 1 on computing the necessary components to perform the FWI (i.e., the forward and adjoint fields [56]) in the truncated domain using the local solver.

We clearly see that Eq. 3 is much smaller than solving the Helmholtz equation in the full domain because $(N_{x_{sub}} \times N_{z_{sub}}) \leq (N_x \times N_z)$. Because of these rapid wavefield computations, FWI in the truncated domain can greatly reduce the turn around time necessary to predict velocity changes around the reservoir. As mentioned above, Eq. 3 depends on the computation of a collection of Green's functions in the full domain once. This is a computationally expensive process since we need to solve $N_s + N_b^{\delta\Omega}$ PDEs for 2D seismic acquisition, where $N_b^{\delta\Omega}$ is the number of gridpoints at $\delta\Omega$ and N_s is the number of sources acquired in the field. For 3D seismic acquisition, the number of gridpoints both at the surface and the boundary of the truncated domain will increase by an order of magnitude. To overcome this computational burden, we propose to exploit the low-rank structure of the full subsurface Green's function to reduce the simulation costs. In the next section, we explain in detail the concept of low rank and its application to reduce the number of PDE solves to extend the local solver for large-scale FWI.

3 Low-rank approximation of the Green's function

The objective of this work is to build an accurate low-rank representation of the Green's function for efficient memory storage and computational time. To do so, we first visualize the structure of the full subsurface Green's function and then study its low-rank structure. Low-rank structure refers to the small number of nonzero singular values or quickly decaying singular values of the underlying matrix. For 2D seismic data acquisition, the full subsurface Green's functions at a single frequency for the sources placed at the surface and on the boundary of the local domain will be a 3D tensor with dimensions N_z, N_x and $(N_s + N_b^{\delta\Omega})$. Since the full subsurface Green's function is a tensor, one way to visualize the decay of its singular values is by matricizing the underlying tensor. Matricization is the process of unfolding an nD tensor into a 2D matrix. In this work, we matricize the 3D tensor by combining the N_z and N_x dimensions along the rows while keeping $(N_s + N_b^{\delta\Omega})$ along the columns. Although there are other matricizations for this 3D tensor, we choose this particular one because it naturally comes from the solution of the wave equation, i.e., each column of this matrix represents a new source experiment.

We find that the full subsurface Green's function exhibits a low-rank structure when organized in the proposed matrix form. To understand the low-rank behavior of the full subsurface Green's function in 2D, we simulate a dataset in the Marmousi model (Fig. 1) by placing the sources at the surface and on the boundary $\delta\Omega$ of the local domain represented by the black box in Fig. 1. We simulate the dataset at 5 Hz using a Ricker wavelet with a central frequency of 12 Hz. We place $N_s = 1361$ sources at the surface and $N_b^{\delta\Omega} = 340$ sources on the boundary of the local domain sampled at 10 m. Given N_s and $N_b^{\delta\Omega}$, we need to simulate $N_s + N_b^{\delta\Omega}$ PDEs in the background model to form the Green's function matrices in conventional local solver implementations for each monochromatic frequency.

From the simulated data, we extract two sets of full subsurface Green's functions. The first set corresponds to the sources at the surface and the second set represents the sources placed on the boundary of the local domain. We then matricize these tensors where the sizes of the matricized tensors are $N_zN_x \times N_s$ and $N_zN_x \times N_b^{\delta\Omega}$, respectively. Here, rows of these matrices represent the solution of the wave equation at each grid point in the subsurface and each column represents a single source experiment with the source either placed at the surface or on the boundary of the local domain. Figure 2 a–c show the matricized full subsurface Green's function at 5 Hz, whereas, Fig. 2 d–f show a column from the full subsurface Green's function. Figure 2 g displays the decay of the singular values of the full subsurface Green's function. We see that the matricized Green's function exhibits low-rank structure, i.e, its singular values decay rapidly. Therefore, we can approximate the full subsurface Green's function using very few singular vectors. To verify this, we reconstruct the full subsurface Green's function keeping only the largest 20% of the singular vectors as shown in Fig. 2b and e. We see that we are able to capture all the coherent energy, which is also validated by the residual plot (Fig. 2c, f). Also, the signal-to-noise ratio (SNR) of the approximated Green's function is 27 dB, where SNR $= -20\log_{10}\frac{\|\text{true}-\text{approximated}\|_2^2}{\|\text{true}\|_2^2}$.

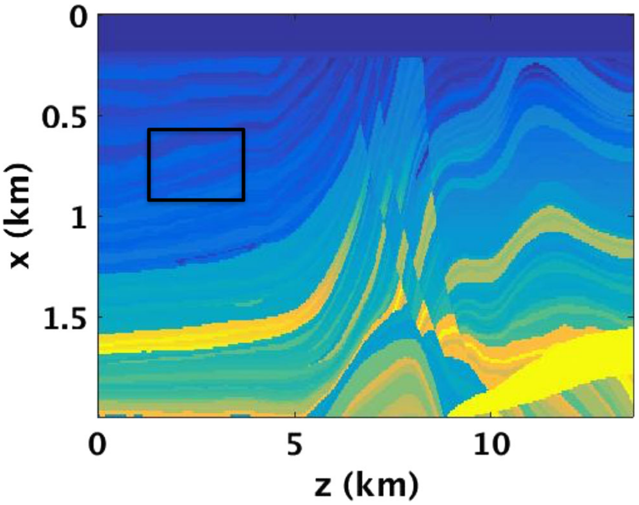


Fig. 1 Marmousi baseline model. We use this model to demonstrate the low-rank representation of the Green's function. The black box represents the local domain Ω

Above, we show a low-rank approximation computed with singular value decomposition (SVD). This is a relatively naive way to compute the low-rank structure as one has to simulate all the required Green's functions in the background model followed by partial singular value decomposition. Unfortunately, this is not a practical solution since computing the SVD can be extremely time-consuming on large-scale models. To overcome this, we propose to test a randomized singular value decomposition approach to approximate the full subsurface Green's function,

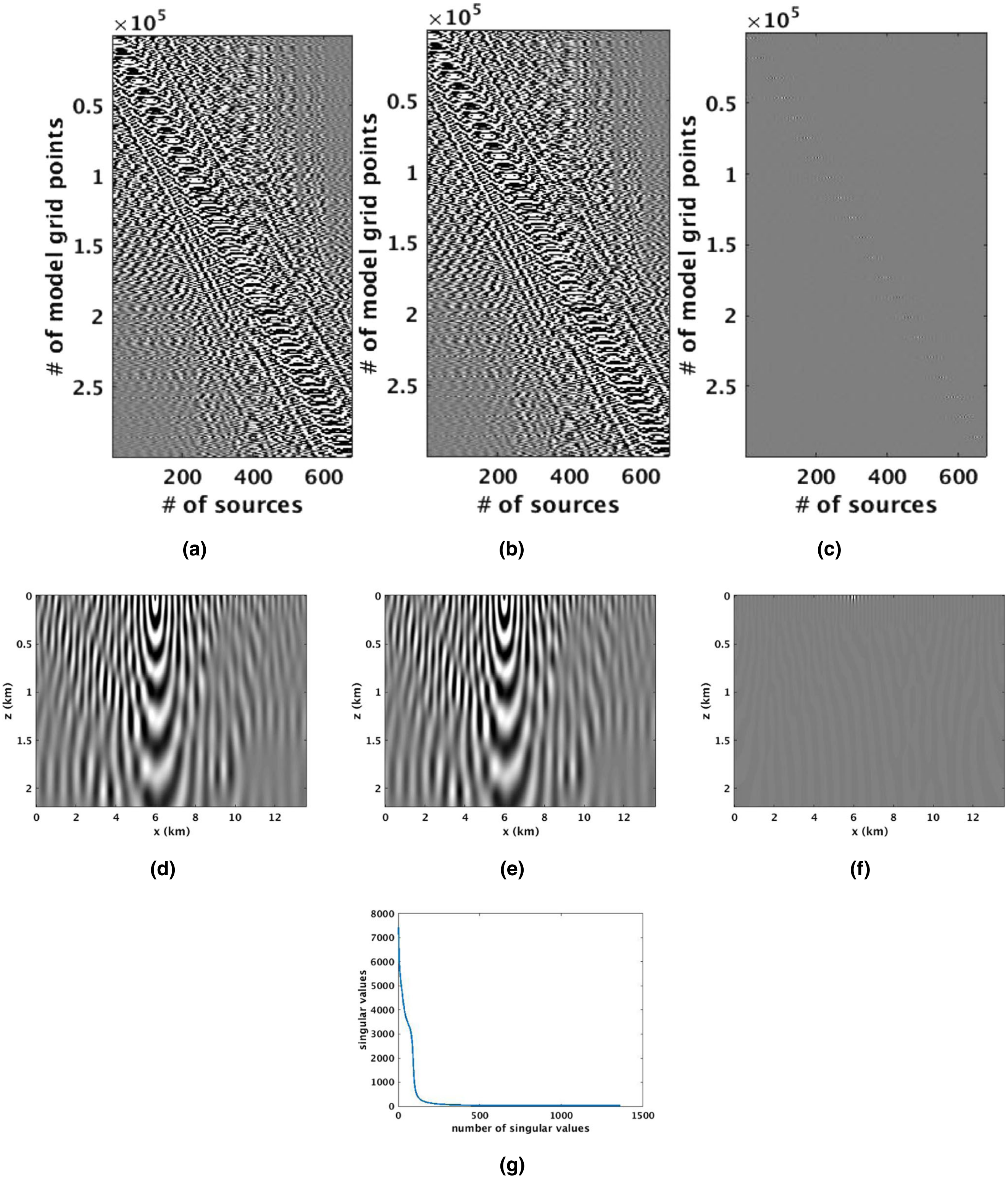


Fig. 2 Green's function simulated on the surface of the local domain. We matricized the 3D Green's function to analyze its singular value decay. **a** True Green's function. **b** Approximated Green's function, where we keep the largest 20% of the singular values. **c** Residual. It is evident from the residual that we are able to recover most of the coherent energy with signal-to-noise ratio of 27 dB. **d**–**f** A single column extracted from **a**–**c**. For display purposes, we display only every second source in the full subsurface Green's function matrices. **g** Singular value decay of the true Green's function in **a**. We clearly see that the singular values decay fast; hence, the underlying Green's function exhibits a low-rank structure

which offers significant speedups and memory saving over classical SVD methods. Algorithm 1 summarizes the process to compute the full subsurface background Green's function in its low-rank SVD form, which is merely the randomized SVD from [18].

3.1 Algorithm 1: randomized SVD of the full subsurface Green's function [18]

1. Input: rank (k) of the monochromatic Green's function, Gaussian random matrix $\mathbf{W} \in \mathbb{C}^{N_s \times k}$, a function that evaluates $\mathbf{F}(\mathbf{m})$
2. Compute
   a. $\mathbf{Y} = \mathbf{F}(\mathbf{m})\mathbf{W}$, the computational cost is that of k PDE solves
   b. $[\mathbf{N}, \mathbf{M}] = \mathrm{qr}(\mathbf{Y})$, where $\mathbf{Y} \in \mathbb{C}^{N_x N_z \times k}$
   c. $\mathbf{Z} = (\mathbf{F}(\mathbf{m})\mathbf{N})^*$, extra k PDEs solves
   d. $[\mathbf{T}, \mathbf{S}, \mathbf{B}] = \mathrm{svd}(\mathbf{Z})$ ($\mathbf{Z} \in \mathbb{C}^{N_x N_z \times k}$ is a small matrix)
   e. $\mathbf{T} \leftarrow \mathbf{NT}$, approximated left singular vectors
3. Output: compressed representation of the full subsurface Green's Function for the sources at the surface, $\mathbf{T} \in \mathbb{C}^{N_x N_z \times k}$, $\mathbf{B} \in \mathbb{C}^{N_s \times k}$, and $\mathbf{S} \in \mathbb{C}^{k \times k}$
4. Repeat steps 1–3 for the sources on the boundary of the local domain with $\mathbf{W} \in \mathbb{C}^{N_b^{\delta\Omega} \times k}$

Here, $\mathbf{F}(\mathbf{m}) = \mathbf{H}(\mathbf{m})^{-1}\mathbf{P}_s^T\mathbf{I}$ represents the full subsurface Green's function for sources placed at the surface or on the boundary of the local domain, which is the solution of the Helmholtz equation. The restriction matrix $\mathbf{P}_s^T \in \mathbb{R}^{N_x N_z \times N_s}$ injects the source wavefields into the grid at the source locations. The matrix $\mathbf{I} \in \mathbb{R}^{N_s \times N_s}$ is the identity matrix. Note that, for the sources on the boundary of the local domain Ω, the sizes of matrices $\mathbf{P}_s^T$, $\mathbf{I}$ are $N_x N_z \times N_b^{\delta\Omega}$ and $N_b^{\delta\Omega} \times N_b^{\delta\Omega}$, respectively. Each column of $\mathbf{W} \in \mathbb{C}^{N_s \times k}$ represents a Gaussian random vector, where k is the rank of the underlying full subsurface Green's function. We choose each of the column vectors of $\mathbf{W}$ in such a way that $\mathbb{E}(\mathbf{w}_i {\mathbf{w}_i}^T)$ equals the identity matrix. Note that the first step of Algorithm 1 represents a simultaneous source experiment where, for each simultaneous source experiment, N_s sources are fired simultaneously with different weights. In total, we perform k simultaneous source experiments each at stages 2a and 2c of Algorithm 1. For large-scale problems, randomized SVD-based low-rank approximation is significantly cheaper than computing an SVD of the full matrix. The overall computational cost of the randomized SVD is of the order $O(N_x^2 \times k)$. As we see in Algorithm 1, instead of solving $(N_s + N_b^{\delta\Omega})$ PDEs using in the full domain as for the classical local solver, we now solve $2k_s + 2k_b$ PDEs to approximate the full subsurface Green's function. Here, k_s and k_b represent the number of simultaneous source experiments to approximate the Green's function at the surface and on the boundary of the local domain, respectively. As long as $(k_s + k_b)$ is significantly smaller than $(N_s + N_b^{\delta\Omega})$, we significantly reduce the required number of full subsurface Green's function in the background model. In addition to the solution of PDEs, we also perform a QR decomposition in step two and singular value decomposition in step four. We want to emphasize that both $\mathbf{Y}$ and $\mathbf{Z}$ in steps 2a and 2c are tall and thin matrices of size $(N_x \times k)$; hence, these steps are computationally cheap to evaluate. Thus, the randomized SVD-based approach is computationally feasible for evaluating the Green's function in the background model for large-scale 2D seismic data problems, where the underlying model dimensions in both the vertical and horizontal directions are of the order $O(10^2 - 10^3)$.

3.2 Verification of the local solver using low-rank approximation

Now that we have the low-rank approximation of the Green's function, the next step is to estimate the accuracy of the local solver with the approximate Green's functions from Algorithm 1. To do this, we compare the forward and adjoint wavefields and the gradient in the full domain $\Omega^c + \Omega$ using the conventional full domain solver and the local domain Ω using the approximate local solver. Figure 3 shows the true baseline Marmousi model, used as the background model $\mathbf{m}_0$, and the perturbation $\delta\mathbf{m}$ in the local domain, which we use to verify the exactness of the local solver. We perform the comparison at 5 Hz, where we use a Ricker wavelet with a central frequency of 12 Hz. We generate the wavefields and gradients in the full domain using the classical method, where we use a finite difference solution of Helmholtz equation in the entire domain Ω.

To compute the forward and adjoint wavefields in the truncated domain, we first approximate the full subsurface Green's function using Algorithm 1. For this example, we choose $k_s = 0.1N_s = 130$ for the sources at the surface and $k_b = 0.1N_b^{\delta\Omega} = 40$ for the sources on the boundary of the local domain. From the approximated full subsurface Green's function, we extract four sets of fully sampled Green's functions, i.e., $\mathbf{u}_0^{(s,r)}$, $\mathbf{u}_0^{\delta\Omega}$, $\mathbf{G}_0^{\delta\Omega}$, and $\mathbf{G}_0^{\delta\Omega+1}$. Appendix 1 demonstrates the process of forming these matrices from the full subsurface Green's function. These four matrices are of size $N_x \times N_x$, $N_b^{\delta\Omega} \times N_s$, $N_b^{\delta\Omega} \times N_b^{\delta\Omega}$, and $N_b^{\delta\Omega} \times (N_b^{\delta\Omega} - 8)$. Next, we compute the forward and adjoint wavefields in the truncated domain as per Eq. 3 where we insert the randomized SVD-based approximated Green's function. Note that, in classical settings of the local solver [58], we need to solve 1701 PDEs using the finite difference solution of wave equation to compute the Green's functions, whereas, we only solve 340, i.e., $2(0.1N_s + 0.1N_b^{\delta\Omega})$ PDEs using the proposed framework.

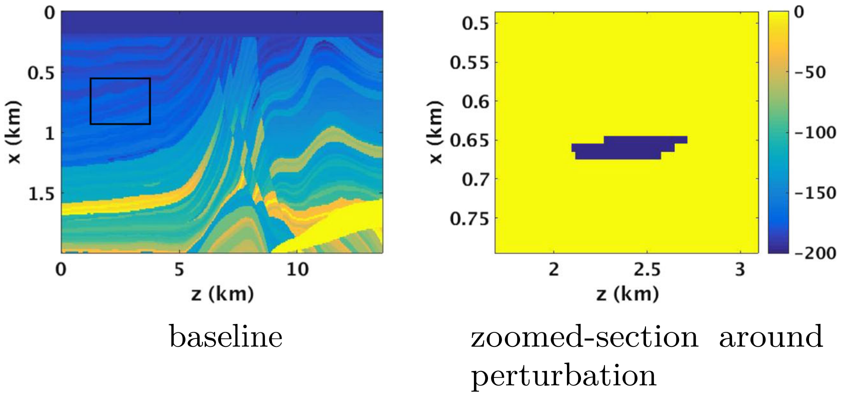


Fig. 3 Baseline model and perturbation for testing the low-rank approximation of Green's function in local solver settings

Figures 4 and 5 compare the real component of the forward and adjoint wavefield simulated in the full domain and the local domain, and Fig. 6 shows the gradient computed via cross-correlating the forward and adjoint wavefields. As pointed out by [58], Eq. 3 preserves all orders of scattering without loss of accuracy between the model perturbation $\delta\mathbf{m}$ and the background model $\mathbf{m}_0$. In our randomized SVD formulation, we see from the residual plot in Fig. 5 that even though we lose some coherent energy, this does not significantly impact the gradient as can be seen in Fig. 6. Hence, the approximate local solver is acceptable to perform the full-waveform inversion in the area of interest.

4 Time-lapse inversion

Having illustrated the computationally efficient randomized SVD process of reducing the number of PDEs to form the full subsurface Green's function, and the accuracy of the approximate local solver using the inexact Green's function, we now move on to perform a time-lapse waveform inversion. We follow the double-difference full-waveform inversion (DDFWI) strategy [11, 36, 59] and focus only on estimating the time-lapse change in the truncated domain Ω. Figure 7 shows the baseline and the time-lapse model we use for the waveform inversion. For this example, we use the full Marmousi velocity model, which is 2 km deep and 12 km wide, sampled at 10 m. The synthetic data contain 600 sources and 600 receivers sampled at 20 m. We use frequencies from 3 to 10 Hz, where the source-signature is a Ricker wavelet with a central frequency of 20 Hz. We use a frequency domain finite difference code [10] to simulate the synthetic data.

To generate the background model for the local solver, we first perform standard full domain FWI over the baseline velocity model, with the same acquisition and inversion settings described above. We use 20 iterations of the LBFGS [51] solver to perform the waveform inversion. We invert for seven frequency batches sequentially; each batch consists of six frequencies sampled at 0.2 Hz. We then use this inverted baseline model to perform the time-lapse waveform inversion in the local domain. To do this, we first approximate the Green's function over the full domain using Algorithm 1, where $k_s = 0.125N_s$ for the sources at the surface and $k_b = 0.125N_b^{\delta\Omega}$ for the sources on the boundary of the local domain. We then use this approximate Green's function to compute the forward and adjoint wavefields followed by the cross-correlation to compute the gradient updates. Figure 8 shows the inverted time-lapse model using the numerically exact local solver [58] and the approximated local solver proposed in this paper. We use 10 iterations for each frequency batch to perform the waveform inversion in the local domain. We see that the proposed method is able to recover the time-lapse change fairly well and the results are comparable with those of the numerically exact local solver. Moreover, the computational cost of

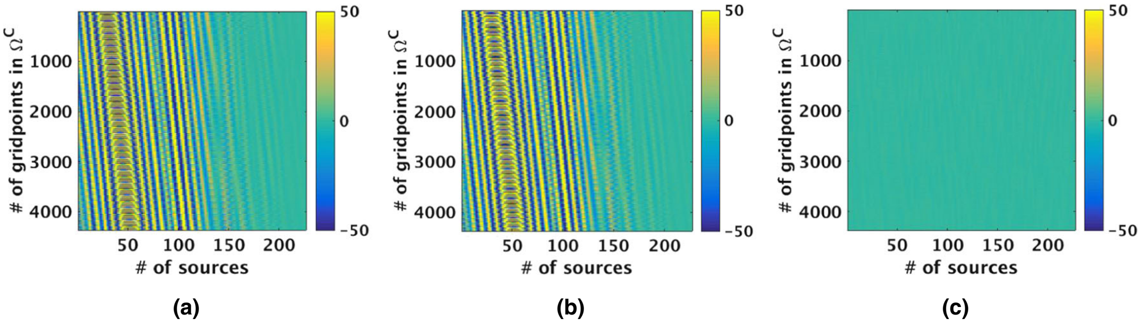


Fig. 4 Real part of forward wavefield in the local domain. **a** True. **b** Approximation using randomized SVD. **c** Residual

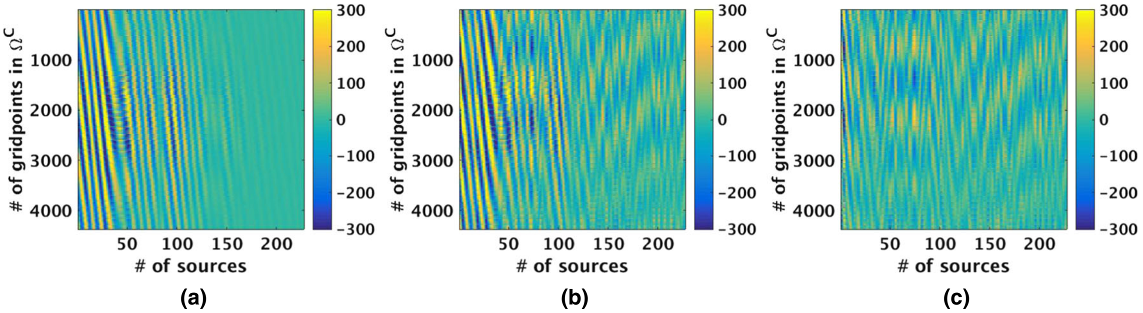


Fig. 5 Real part of adjoint wavefield computed in the local domain. **a** True. **b** Approximation using randomized SVD. **c** Residual

evaluating the Green's function in the proposed approach is four times smaller than the exact local solver approach. This shows that the randomized SVD approach enables the computation of the Green's function cheaply in the full domain, thus mitigating the computational bottleneck of the numerically exact local solver to perform waveform-inversion in the local domain.

Although we reduce the computational cost of the local solver drastically, the inverted time-lapse result shows more artifacts, which we believe is due to the fact that we are using the LBFGS solver with inexact gradient information at each iteration. If the gradients are approximated at every iteration, then the LBFGS solver might break down faster, thus resulting in inaccurate update directions. We believe conjugate-gradient (and especially gradient descent) might be more stable for this kind of scenario. We could also mitigate these problems by adding a regularization term in the approximated local solver to further stabilize the inversion process. Both of these topics are subjects of future research.

5 Extension to 3D FWI

Motivated by the success of 2D target-oriented inversion using the approximated local solver, the next step is to extend it to 3D full-waveform inversion, where the computation of background Green's functions becomes prohibitively expensive. Again, one way to circumvent this computational bottleneck is by using the randomized SVD approach of Algorithm 1 to approximate the Green's function in 3D. Although a randomized SVD-based framework is computationally efficient to estimate the Green's function, one needs to solve $2k$ PDEs as shown in Algorithm 1. For large-scale 3D problems, k can easily be in the range of tens of thousands. Note that we need to solve an additional $2k$ PDEs to approximate the Green's function for the sources on the boundary of the local domain. Thus, the randomized SVD-based framework will also be computationally demanding for performing 3D time-lapse FWI in the local domain, even if k is much smaller than the number of sources $(N_{x_{src}}N_{y_{src}} + N_b^{\delta\Omega})$.

If we look carefully at step 2a of Algorithm 1, then we immediately see that we are simulating k simultaneous source experiments, where we multiply source matrix $\mathbf{I} \in \mathbb{R}^{N_s \times N_s}$ with the Gaussian random matrix $\mathbf{W} \in \mathbb{R}^{N_s \times k}$. This will create blended coherent noise in each shot experiment. Given the simultaneous sources at the first step of Algorithm 1, we design a computationally efficient source-separation framework, where the cost of source separation is a fraction of the cost of solving the extra k PDEs at step 2c in Algorithm 1. Hence, for 3D local waveform inversion, we use a rank-minimization–based source-separation technique to remove the coherent interference to recover the fully sampled Green's function in the background model. This reduces the number of PDE solves from $2(k_s + k_b)$ to

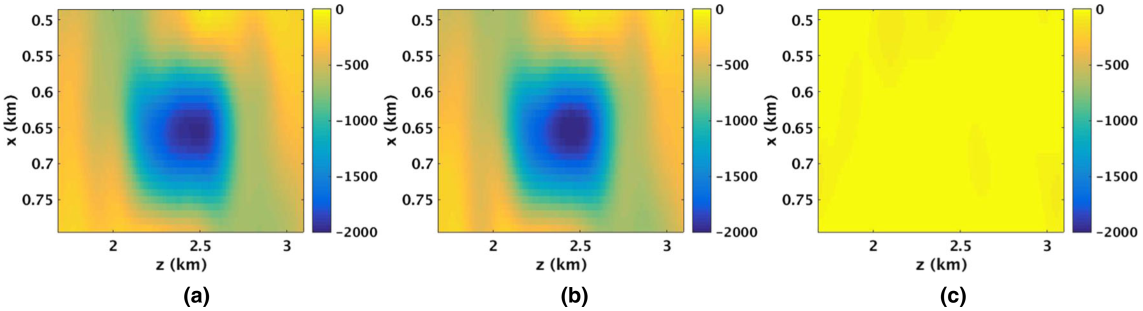


Fig. 6 Gradient comparison in the local domain. **a** True. **b** Approximation using randomized SVD. **c** Residual

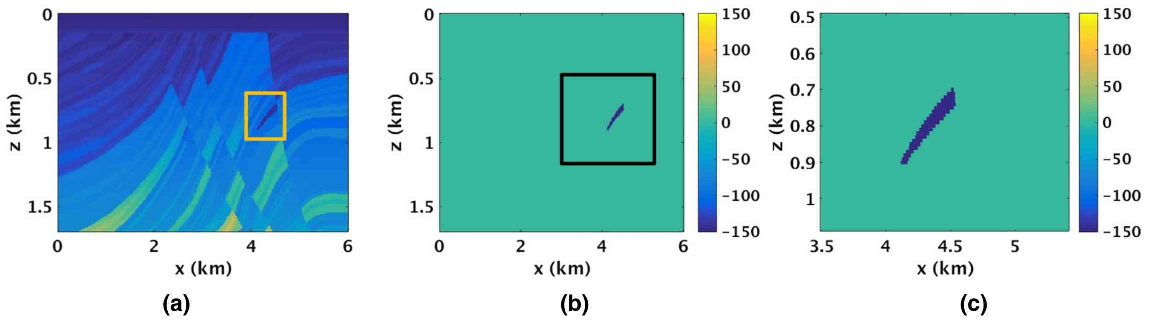


Fig. 7 Time-lapse inversion. **a** Baseline and **b** time-lapse section. The orange box in **a** represents the time-lapse region. **c** Local domain around time-lapse change (black box in **c**), which is used to illustrate the advantages of the proposed approach

$(k_s + k_b)$ for approximating the Green's function matrices for the sources at the surface and on the boundary of the local domain.

For 3D seismic data, we only do $k_s + k_b$ simultaneous source simulations at step 2a of Algorithm 1 to generate the full subsurface simultaneous Green's function for the sources at the surface and on the boundary of the local domain. From this full subsurface simultaneous Green's function matrix, we then extract two sub-matrices, one for source locations on the surface and one for source locations on the boundary. Finally, we perform the source separation on each of these sub-matrices independently. Thus, we only need to solve $(k_s + k_b)$ PDEs to approximate the Green's function matrices to solve the local domain wave equation system defined in Eq. 3.

Various methodologies have been proposed to extract the seismic data from the blended data [1, 3, 41, 49, 57]. Here, we rely on the compressed sensing (CS) framework [9, 13] that offers three fundamental principles for successful reconstruction of the original signal sampled at the sub-Nyquist rate. Specifically, we use the rank-minimization–based framework to approximate Green's function matrices, which is a natural extension of CS ideas for data volumes organized as matrices. Interested readers can look into [19, 27] for extensive details on these principles and its usage for seismic data interpolation and/or deblending. In the next section, we briefly describe the three fundamental principles for successful reconstruction of the Green's function from relatively few simultaneous sources using the rank-minimization–based framework.

5.1 Low-rank structure of Green's function

The first principle utilizes the prior knowledge that the underlying fully sampled signal of interest should exhibit a fast decay of its singular values and can thus be well approximated by a low-rank matrix. To understand the first principle, we visualize the low-rank structure of the Green's function organized as a matrix for the sources placed at both the surface and on the boundary of the local domain, i.e, $\delta\Omega$. To demonstrate the low-rank behavior, we simulate the fully sampled sequential source Green's function using the SEG/EAGE 3D overthrust model (Fig. 9). The dimension of the model is 5 km × 20 km × 20 km and is discretized on a 25 m × 25 m × 25 m grid. Here, we analyze the low-rank properties of the Green's function for two scenarios. In the first scenario, we place the sources and receivers at the surface spaced by 100 m and 50 m, respectively, along inline and crossline directions. In the second scenario, we place sources and receivers on the boundary of the local domain. For both scenarios, we use a Ricker wavelet with peak frequency of 10 Hz and simulate the data using a finite difference time-domain modeling code [29]. We then apply a temporal-Fourier transform

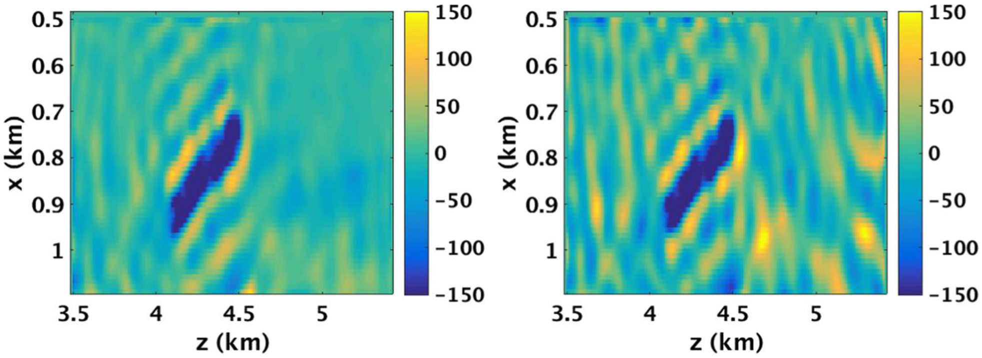


Fig. 8 Time-lapse waveform inversion results using the double-difference approach [11, 59]. **a** Using the numerically exact local solver [58]. **b** Using the proposed randomized SVD-based local solver

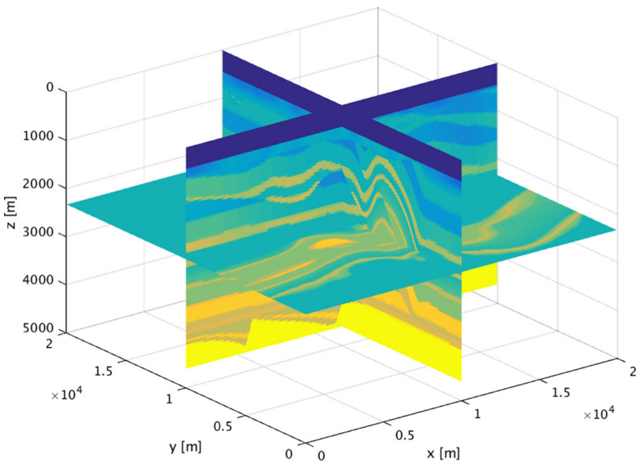


Fig. 9 SEG/EAGE overthrust model

to extract the Green's function matrices in the frequency domain.

From the simulated data, we extract three sets of fully sampled Green's functions, i.e., $\mathbf{u}_0^{(s,r)}$, $\mathbf{G}_0^{\delta\Omega}$, and $\mathbf{G}_0^{\delta\Omega+1}$ at 5 Hz. Here, each monochromatic Green's function restricted to the surface and the boundary of the local domain will be a 4D tensor of size $N_{x_{src}}$, $N_{y_{src}}$, $N_{x_{rec}}$, and $N_{y_{rec}}$. As mentioned before, to analyze the low-rank behavior of a tensor, we need to unfold it into a matrix, since the concept of singular value decomposition (SVD) is for matrices only. For these 4D Green's functions, we follow the matricization strategy proposed by [25, 53] which show that grouping $N_{x_{src}}, N_{y_{src}}$, i.e., placing both the source coordinates along the columns (Fig. 10a), results in a higher rank or slow decay of singular values as shown in Fig. 10 c. However, grouping the $N_{x_{src}}, N_{x_{rec}}$ along the rows and $N_{y_{src}}, N_{y_{rec}}$ along the columns (Fig. 10b) results in a matrix with fast decay of the singular values (Fig. 10c).

5.2 Effect of blending on the low-rank structure

The second principle of the compressed sensing-based source-separation framework is based upon a sampling scheme that breaks the underlying structure—i.e., increases the rank or slows down the decay of the singular values of the original signal. In our case, this translates to finding a simultaneous source scheme that increases the rank of the underlying fully sampled matrix in the transform domain. To understand this, we analyze the effect of blending on the low-rank properties of the Green's function in the transform domain. To mimic the simultaneous source experiment, we multiply the $N_{x_{src}}, N_{y_{src}}$ matricized Green's function (Fig. 10a) with the Gaussian random matrix $\mathbf{W}$ of size $N_{x_{src}} N_{y_{src}} \times k$, where k represents the number of simultaneous source experiments. In this example, we choose $k = 0.1 * N_{x_{src}} N_{y_{src}} \approx 1000$. Since we want to analyze the effect of blending on the decay of the singular values in the transform domain, we create a pseudo deblended Green's function (Fig. 11) in the $N_{x_{src}}, N_{y_{src}}$ and $N_{x_{src}}, N_{x_{rec}}$ matricization, respectively.

We further plot the decay of the singular values after pseudo deblending in the $N_{x_{src}}, N_{y_{src}}$ and $N_{x_{src}}, N_{x_{rec}}$ matricization as shown in Fig. 11. We see that blending destroys the continuity of the waveforms in both matricizations, which results in slower decay of the singular values; thus, the rank increases dramatically. It is clear that although for both parameterizations the decay of singular values is slower than without the simultaneous sources, the decay of singular values is significantly faster for the $N_{x_{src}}, N_{y_{src}}$ matricization than it is for the $N_{x_{src}}, N_{x_{rec}}$ matricization; in any case, we would use the latter matricization because this is the matricization in which the underlying data are sparse.

5.3 Rank-minimization framework

To reconstruct the Green's function matrix from blended measurements, we use the rank-minimization framework for simultaneous source separation proposed by [26], where the underlying assumption is that the fully sampled Green's function matrix exhibits low-rank structure and the blending process increases the rank of the Green's function matrix in some transform domain. Under this assumption, the source separation problem is to find the fully sampled Green's function matrix of lowest possible rank that agrees with the simultaneous source experiment observations.

For a low-rank matrix $\mathbf{X}$ in $\mathbb{C}^{n\times m}$ and a linear measurement operator $\mathcal{A}$ that maps from $\mathbb{C}^{n\times m} \rightarrow \mathbb{C}^{p}$ with $p \ll n \times m$, the rank-minimization problem involves solving the following problem for $\mathcal{A}$, up to a given tolerance ϵ:

$$\underset{\mathbf{X}}{\text{minimize}} \quad \operatorname{rank}(\mathbf{X}) \quad \text{subject to} \quad \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2 \le \epsilon, \tag{4}$$

where $\mathbf{b}$ is a set of blended measurements. Note that, for a 3D seismic data acquisition, $\mathcal{A}$ represents the Gaussian random matrix $\mathbf{W}$, $m = N_{x_{src}} \times N_{y_{rec}}$, $n = N_{y_{src}} \times N_{x_{rec}}$, b is the simultaneous source Green's function matrix extracted from $\mathbf{Y}$, which corresponds to the sources at the surface or on the boundary of the local domain, and $p = N_{y_{src}} \times N_{x_{rec}} \times k$. Appendix 2 explains a computationally efficient framework to solve (4) efficiently for large-scale seismic data acquisition.

Finally, we propose a two-step strategy to efficiently compute the Green's function in the background model to enable the numerically exact local solver for large-scale 3D seismic data problems. The first step involves solving the partial differential wave equation (PDEs) k times using the simultaneous sources placed at the surface of the domain, i.e., step 2a of Algorithm 1. The second step involves the source-separation procedure, i.e., deblending,

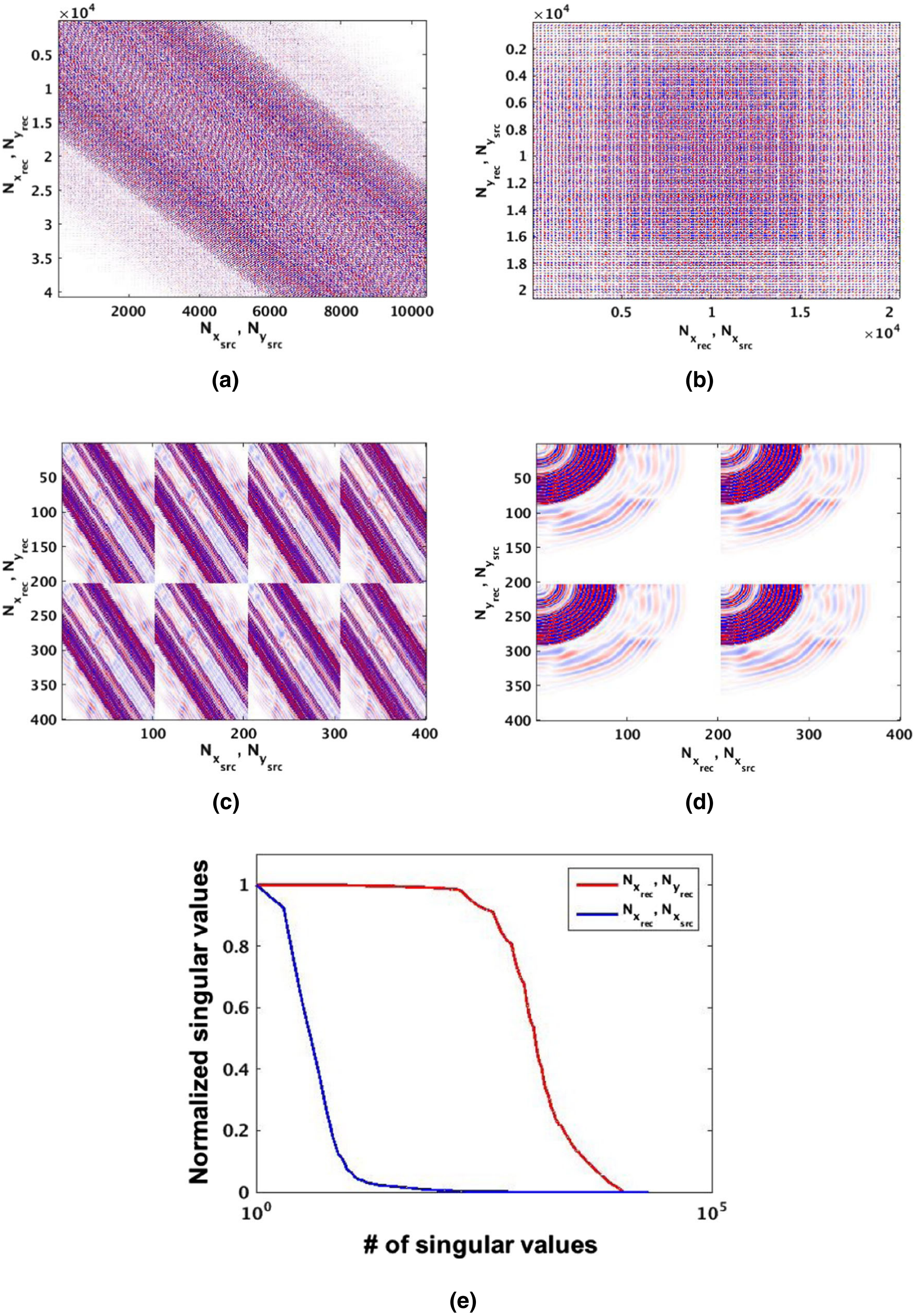


Fig. 10 Matricized Green's function simulated using the SEG/EAGE overthrust model at 5 Hz for sources placed at surface. **a** $N_{x_{src}}, N_{y_{src}}$ and **b** $N_{x_{src}}, N_{x_{rec}}$ matricization. **c** and **d** Zoom sections of **a** and **b** at the upper left corner, respectively. **e** Singular value decay. The red and blue curves represent the $N_{x_{src}}, N_{y_{src}}$ and the $N_{x_{src}}, N_{x_{rec}}$ matricizations, respectively. We see that singular values of the fully sampled Green's function decay faster for matricization **b** compared with **a**

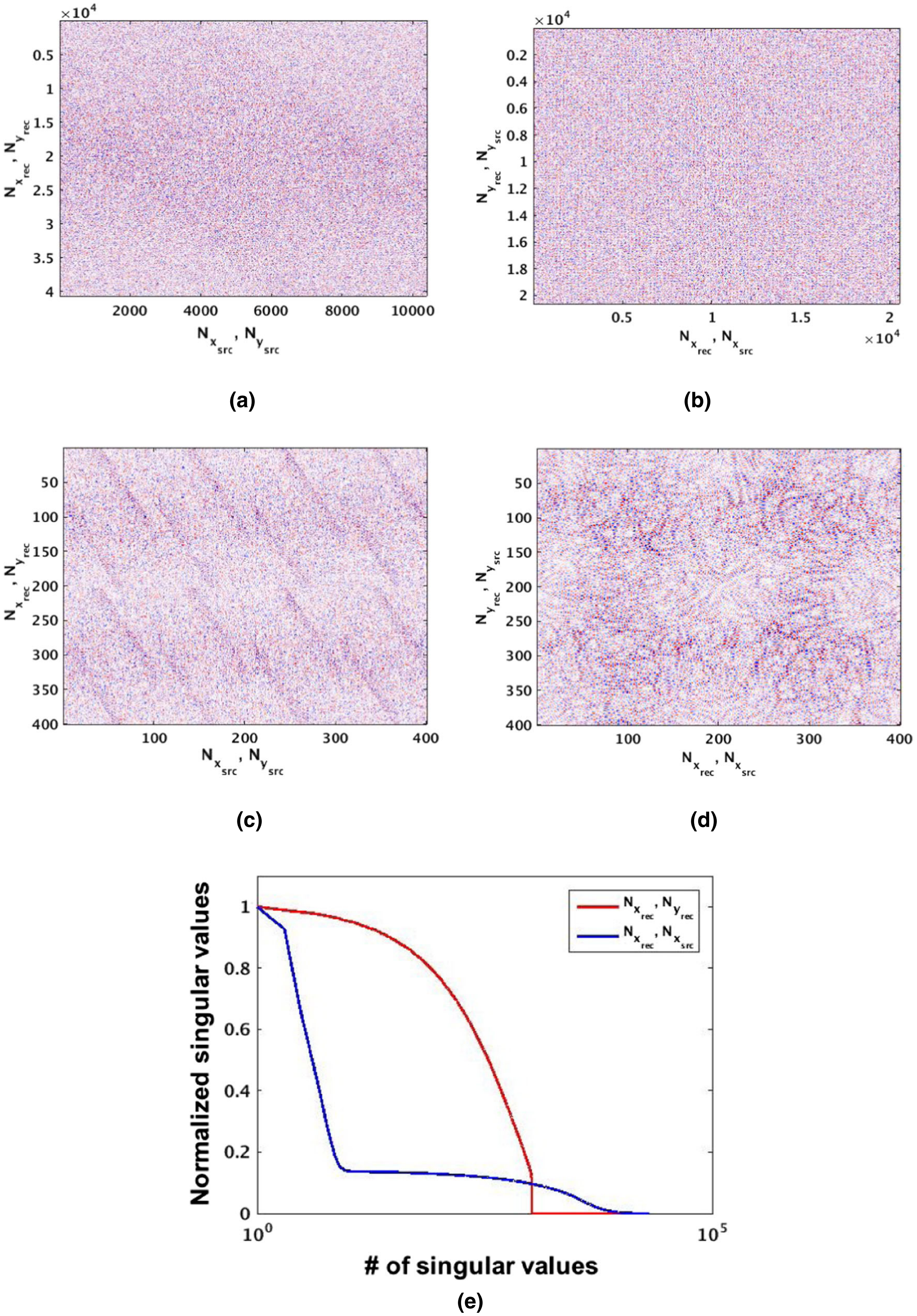
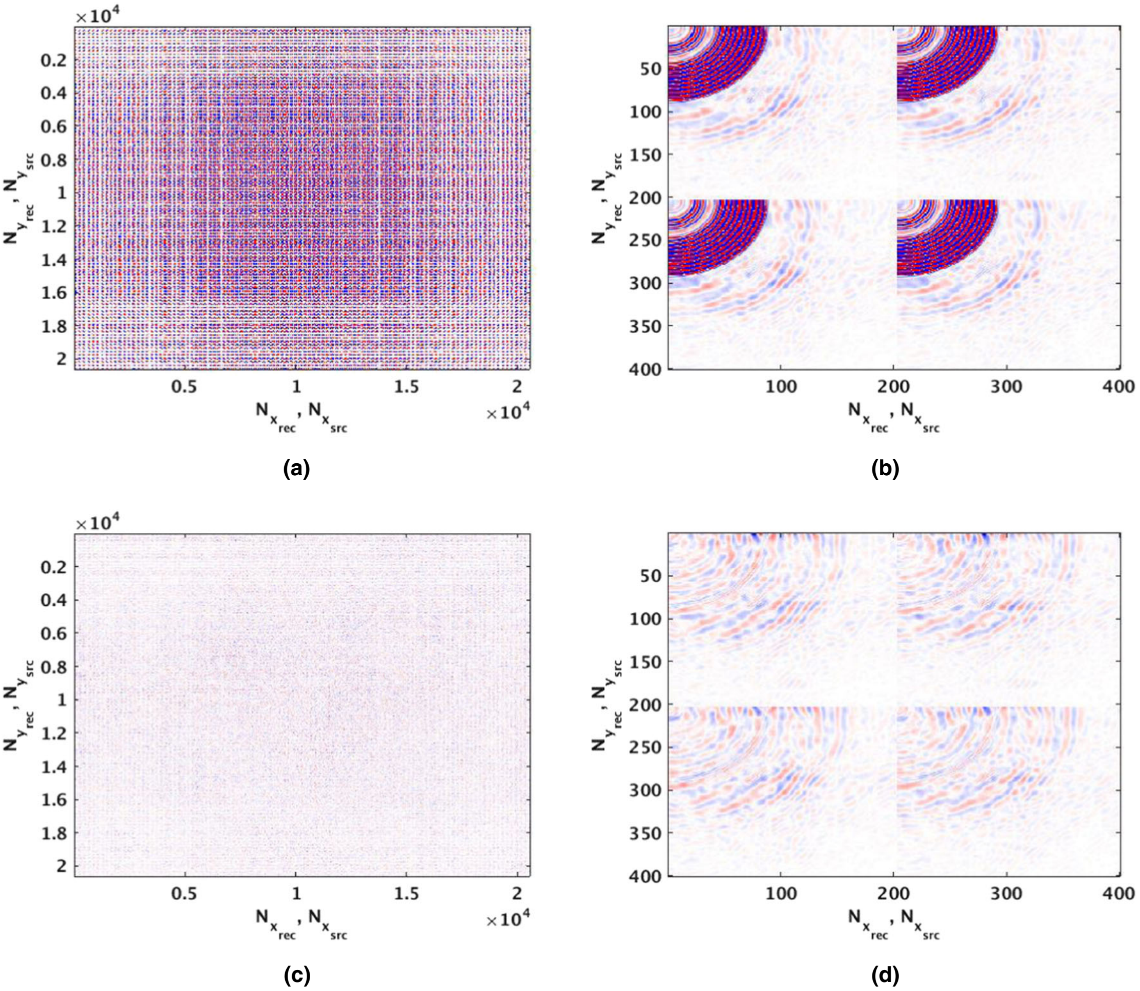


Fig. 11 Pseudo deblended Green's function matrices for sources placed at surface in the **a** $N_{x_{src}}, N_{y_{src}}$ and **b** $N_{x_{src}}, N_{x_{rec}}$ matricizations. **c** and **d** Zoom sections of **a** and **b** at the upper left corner, respectively. **e** Singular value decay. The red and blue curves represent the $N_{x_{src}}, N_{y_{src}}$ and the $N_{x_{src}}, N_{x_{rec}}$ matricizations, respectively. We see that the simultaneous source acquisition slows down the decay of the singular values of $N_{x_{src}}, N_{y_{src}}$ matricization compared with $N_{x_{src}}, N_{y_{rec}}$ matricization

where we recover the fully sampled Green's function, which corresponds to the sources placed at the surface, from the simultaneous source experiment using the rank-minimization framework of Eq. 4. Now, we only need to simulate $k << N_{x_{src}} N_{y_{src}}$ PDEs followed by the source-separation framework, thus avoiding the extra k PDE solves in step 2c of Algorithm 1 to approximate the Green's function. Note that we need to repeat the abovementioned strategy for the simultaneous sources on the boundary of the local domain to approximate $\mathbf{G}_0^{\delta\Omega}$ and $\mathbf{G}_0^{\delta\Omega+1}$ matrices, which will incur the cost of solving an extra k PDEs. Algorithm 2 outlines the rank-minimization–based source-separation framework to approximate the Green's function to allow for the extension of the local solver to large-scale 3D problems.

To further demonstrate the effectiveness of the rank-minimization–based framework to approximate the Green's function, we perform the source separation on the blended Green's function matrices as shown in Fig. 11. The fully sampled Green's function at the surface consists of 102 × 102 sources and 202 × 202 receivers. Since $k = 1000$, the number of PDEs solve is 10 times smaller than the method in [58], where we need to solve 10,404 PDEs conventionally to form the Green's function matrices at the surface. Note that, in the conventional method, we also need to solve thousands of extra PDEs for evaluating the Green's function matrices on the boundary of the local domain. However, in our framework, we only need to solve an extra k PDEs. Figures 12 and 13 show the reconstructed fully sampled Green's function after source separation. We are able to deblend all the Green's function with signal-to-noise ratio (SNR) of ≈ 27 dB and recover most of the coherent energy, which is further supported by the difference plots (Figs. 12c, d and 13c, d). This shows that, using ideas from compressed sensing and a rank-minimization–based interpolation framework, we can mitigate the computational bottleneck of the local solver to perform 3D target-oriented full-waveform inversion. Finally, in Table 1,



(a)
(b)
(c)
(d)

Fig. 12 Recovered fully sampled Green's function using rank-minimization–based framework for sources placed at surface. **a** and **c** After source separation and difference. **b** and **d** Zoom sections from the upper left corner from **a** and **c**. Note that we subtract **a** and Fig. 10 b to generate **c**

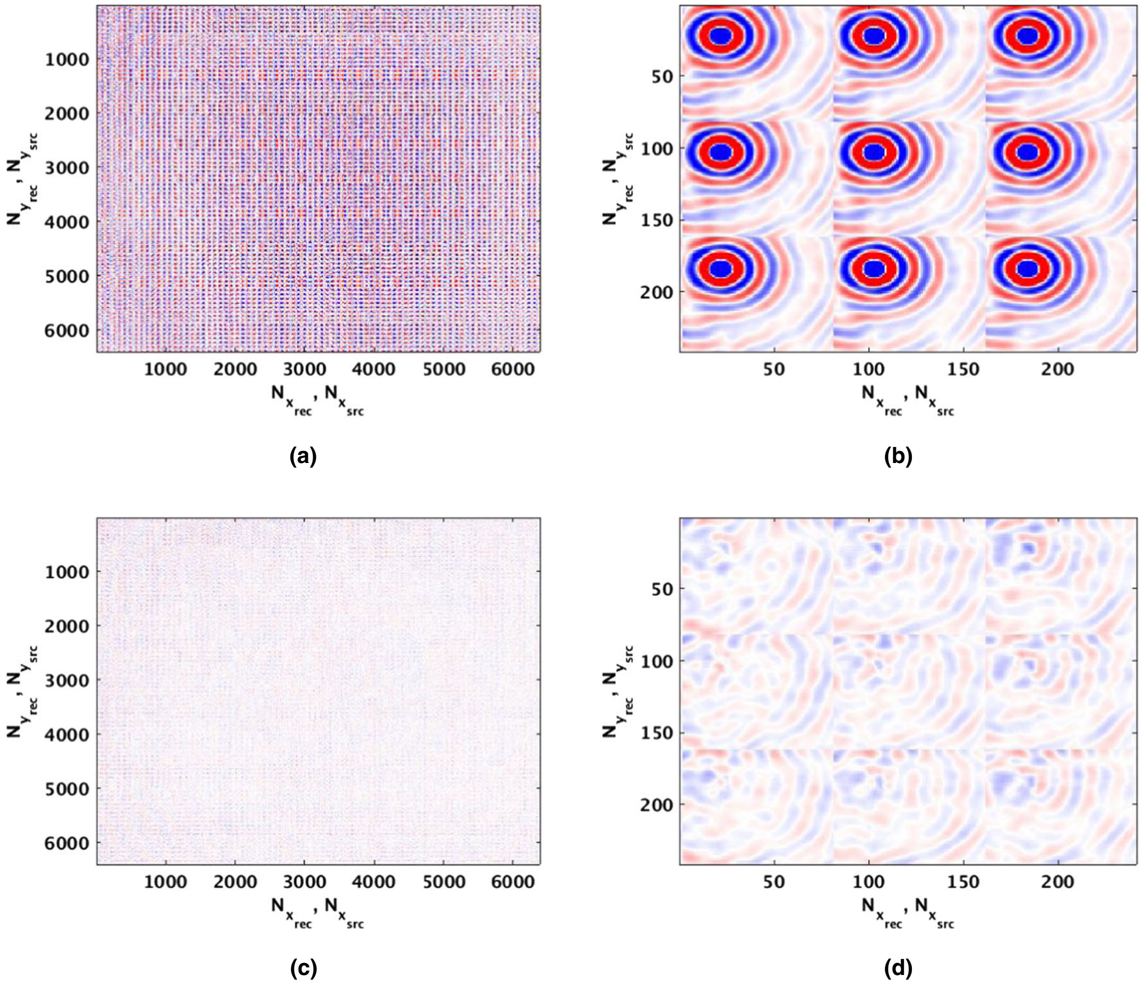


Fig. 13 Recovered fully sampled Green's function using the rank-minimization–based framework for sources placed on the boundary of the local domain. **a** and **c** After source separation and difference. **b** and **d** Zoom sections from the upper left corner from **a** and **c**

we compare the computational cost (in terms of the number of PDEs solve) of full domain FWI, numerically exact local solver, Algorithm 1, and Algorithm 2, respectively.

5.3.1 Algorithm 2: rank-minimization–based source-separation framework

1. Input: $\mathbf{W} \in \mathbb{C}^{N_{xsrc}N_{ysrc}\times k}$, $\mathbf{F}(\mathbf{m})$ (as explained in Algorithm 1)
2. Compute
   a. $\mathbf{Y} = \mathbf{F}(\mathbf{m})\mathbf{W}$, the computational cost is k PDEs, and $\mathbf{Y} \in \mathbb{C}^{N_x N_y N_z \times k}$
   b. Extract simultaneous source Green's function matrices from $\mathbf{Y}$ corresponding to sources at the surface.
   c. Organize this matrix as a vector $\mathbf{b}$.
   d. Solve $\min_{\mathbf{L},\mathbf{R}} \quad \|\mathcal{A}(\mathbf{L}\mathbf{R}')-\mathbf{b}\|_2$ s.t. $\frac{1}{2}\|\mathbf{L};\,\mathbf{R}\|_F^2 \le \tau$, Appendix 2 gives details.
3. Output: Low-rank factorized form of the compressed full subsurface Green's Function, $\mathbf{L} \in \mathbb{C}^{N_{xsrc}N_{xrec}\times k}$, $\mathbf{R} \in \mathbb{C}^{N_{ysrc}N_{yrec}\times k}$
4. Repeat steps 1–3 for the sources placed on the boundary of the local domain to get the corresponding approximated Green's function matrices.

Table 1 Comparison of the computational cost (in terms of the number of PDEs solve) of full domain FWI, numerically exact local solver, Algorithm 1, and Algorithm 2, respectively

	Full domain FWI	Exact local solver	Algorithm 1	Algorithm 2
Number of PDE solves	$2N_s$	$N_s + N_b^{\delta\Omega}$	$2k_s + 2k_b$	$k_s + k_b$

Here, N_s represents the number of sources at the surface, $N_b^{\delta\Omega}$ is the number of points at the boundary of the local domain $\delta\Omega$, and k_s and k_b are the number of simultaneous source experiments at the surface and on the boundary of the local domain, respectively

6 Discussion

The main computational bottleneck of the numerically exact local solver is to simulate the Green's function at the surface and on the boundary of the local domain in the background velocity model. Once we simulate all the Green's functions, performing waveform inversion is computationally cheap to invert for the subsurface parameters of interest. The obvious drawback of this approach is that the required number of wave equation solves to simulate the Green's function grows linearly with the number of grid points; thus, the framework quickly becomes prohibitively expensive for any realistically sized models. To address this, we propose to exploit the low-rank structure of the full subsurface Green's function by using techniques from randomized linear algebra. We show that the simulation cost for monochromatic Green's function is dominated by the rank instead of by the grid points at the surface and on the boundary of the local domain. This observation makes the local solver a computationally feasible and scalable framework for performing target-oriented waveform inversion. As long as the rank is smaller, which is the case for low- to mid-range frequencies, our approach outperforms the conventional numerically exact local solver in terms of computational speed.

Although the low-rank factorization scheme circumvents the computational bottleneck, the obvious question is how to choose the rank for the different monochromatic Green's function. One possible solution is to estimate the rank at the lower and higher end of the spectrum of interest, perform linear interpolation between these two values, and assign rank values to the intermediate frequencies. Estimating the rank value for the lower end of the spectrum is computationally feasible since we can reduce the grid density for the lower frequency and analyze the singular value decay. However, for the higher end of the spectrum, analyzing the decay of the singular values will again become computationally demanding. This is still an open avenue of research.

7 Conclusions

The numerically exact local solver opens new avenues to perform target-oriented full-waveform inversion for time-lapse seismic data acquisition as well as other situations in which only a part of the model is of interest. The central idea is to restrict the computation of partial differential equations to be inside the local domain of interest. Even though the experimental demonstration in seismic literature has shown the benefits of an exact local solver, it requires the computation of Green's functions in the background velocity model. Moreover, the number of wave-equation solves for the Green's function depends upon the number of grid points at the surface and on the boundary of the local domain. This aspect of local solver makes it computationally demanding to perform target-oriented FWI, especially for large-scale 3D full-waveform inversion.

In this work, by exploiting the low-rank structure of the full subsurface Green's function, we circumvent the major computational cost involved in the local solver. Our approach uses probing techniques from the randomized linear algebra to obtain the full subsurface Green's function in a low-rank factorized form. The computational cost of approximating the Green's function using probing techniques depends upon the rank of the Green's function and not on the grid points at the surface and on the boundary of the local domain. As long as this rank is small, which is usually the case for low- to mid-range frequencies, the probing techniques enable the application of the approximate local solver for large-scale seismic data acquisition. Time-lapse inversion results on the 2D Marmousi model demonstrate that we gain a factor of four in the computational time with little to no compromise in accuracy using the proposed low-rank factorization approach to approximate the Green's function.

To extend this framework to 3D full-waveform inversion, we further proposed to combine rank-minimization–based source-separation framework with the randomized SVD to eliminate the need to solve extra PDEs at step 2c in Algorithm 1. This new framework exploits the fact that blending increases the rank of the underlying seismic data, which is low-rank in its deblended form. Using a stylized example from the 3D overthrust model, we demonstrate that we gain a factor of ten in computation time compared with the conventional exact local solver by approximating the Green's function.

Funding information The first author would like to thank the member organizations of the SINBAD II project and the SINBAD Consortium for supporting this work. This study was supported by Chevron, NSERC, InnovateNL, and the Hibernia Management and Development Corporation.

Appendix 1

To perform FWI in the truncated domain, we need to compute the forward and adjoint wavefields. In order to do so, we need to first simulate the Green's function at the surface and on the boundary nodes in the background model. In this section, we explain the process of extracting the Green's function matrices in the background model to efficiently solve the PDEs, i.e., Eq. 3 in the local domain. For a 2D background velocity model, the dimensions of the full subsurface Green's function responses are $N_zN_x \times N_s$

and $N_z N_x \times N_b^{\delta\Omega}$ from which we extract the four desired sets of Green's function matrices, i.e., $\mathbf{u}_0^{(s,r)}$, $\mathbf{u}_0^{\delta\Omega}$, $\mathbf{G}_0^{\delta\Omega+1}$, and $\mathbf{G}_0^{\delta\Omega}$. The matrix $\mathbf{u}_0^{(s,r)}$ represents the Green's function matrix computed in the background model between the sources and receivers at the surface, which we convolve with the source wavelet while computing the data residual during the inversion process. Moreover, we convolve $\mathbf{u}_0^{\delta\Omega}$ with the source wavelet (data residual) while computing the forward (adjoint) wavefields using Eq. 3. The matrices $\mathbf{u}_0^{(s,r)}$ and $\mathbf{u}_0^{\delta\Omega}$ are easy to extract from the full subsurface Green's function matrix via taking out the rows corresponding to the locations of the receivers at the surface and the location of the nodes on the boundary of the local domain, respectively. To form the other Green's function matrices, we follow a node numbering scheme in the truncated domain where the nodes are numbered in a counter-clockwise inward spiraling fashion as shown in Fig. 14 a.

Under this numbering scheme, we obtain the matrix $\mathbf{u}_0^{\delta\Omega}$ from the full subsurface Green's function simulated using the sources at the surface of the model. Again, we select the rows corresponding to the locations of the boundary nodes $\delta\Omega$ for each source experiment. We illustrate the matrix structure of the Green's function matrices $\mathbf{G}_0^{\delta\Omega}$ and $\mathbf{G}_0^{\delta\Omega+1}$ used in the second block row of Eq. 3 by using the 5 × 5 node example from Fig. 14 a.

The foundation of the local solver is the equation for the scattered field. It is an essential component of the local solver in Eq. 3 (i.e., it forms the second block row) and also gives the ability to propagate the local wavefield solution to the receiver locations. This scattered field equation is derived in Appendix 1 of the work of [58] and the final result is restated here for convenience

$$\sum \frac{1}{h^2}\left(u^{\delta\Omega}\left(G_0^{\delta\Omega+1}-G_0^{\delta\Omega}\right) -G_0^{\delta\Omega}\left(u^{\delta\Omega+1}-u^{\delta\Omega}\right)\right) = -\delta u(i,\omega), \quad i \in \delta\Omega \cup \Omega^c, \quad (5)$$

where all the quantities are scalars, h is the grid spacing, and the summation goes all around the boundary but does not include the corner nodes as we will illustrate below. Equation 5 closely resembles a discretization of Green's third identity. We simplify Eq. 5 by removing the common term $u^{\delta\Omega}G_0^{\delta\Omega}$ to get

$$\sum \frac{1}{h^2}\left(u^{\delta\Omega}G_0^{\delta\Omega+1} - G_0^{\delta\Omega}u^{\delta\Omega+1}\right) = -\delta u(i,\omega), \quad i \in \delta\Omega \cup \Omega^c. \qquad (6)$$

The second block row of Eq. 3 evaluates (6) for each boundary node i on $\delta\Omega$, with $i \in \{1, ..., 16\}$ in this 5 × 5 example. The matrices $\mathbf{G}_0^{\delta\Omega+1}$ and $\mathbf{G}_0^{\delta\Omega}$ therefore have 16 rows in this example. Figure 14 b illustrates which Green's function combinations are required when the boundary summation (6) is evaluated around $\delta\Omega$ for the case when $i = 1$. Evaluating (6) gives

$$\begin{aligned} &[u(2)G(i,17) - u(17)G(i,2)] + [u(3)G(i,18) \\ &-u(18)G(i,3)] + [u(4)G(i,19) - u(19)G(i,4)] \\ &+[u(6)G(i,19) - u(19)G(i,6)] + [u(7)G(i,20) \\ &-u(20)G(i,7)] + [u(8)G(i,21) - u(21)G(i,8)] \\ &+[u(10)G(i,21) - u(21)G(i,10)] + [u(11)G(i,22) \\ &-u(22)G(i,11)] + [u(12)G(i,23) - u(23)G(i,12)] \\ &+[u(14)G(i,23) - u(23)G(i,14)] + [u(15)G(i,24) \\ &-u(24)G(i,15)] + [u(16)G(i,17) - u(17)G(i,16)] \\ &= -\delta u(i), \end{aligned} \qquad (7)$$

where each term in square brackets represents a contribution from a single box in Fig. 14 b. Equation 7 is the multiplication of vectors $\mathbf{u}^{\delta\Omega}$ and $\mathbf{u}^{\delta\Omega+1}$ with the first row of matrices $\mathbf{G}_0^{\delta\Omega+1}$ and $\mathbf{G}_0^{\delta\Omega}$ in Eq. 3. We now show the first

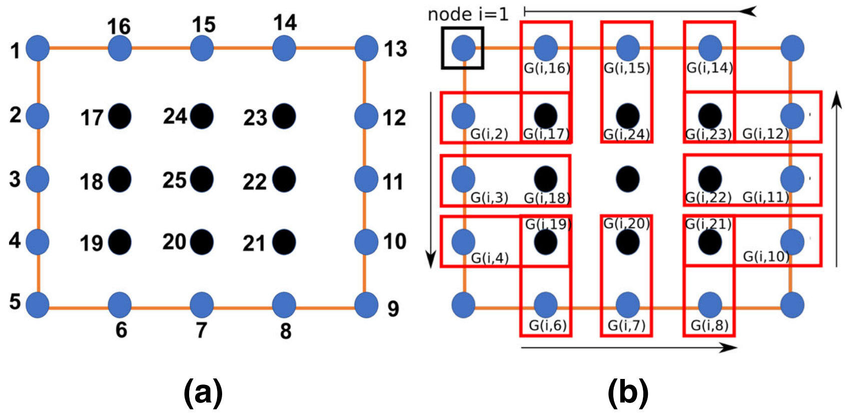


Fig. 14 A 5 × 5 model representation of the local domain Ω. This truncated domain is used to demonstrate the computation of Green's function matrices $\mathbf{G}_0^{\delta\Omega}$ and $\mathbf{G}_0^{\delta\Omega+1}$ used in Eq. 3 to evaluate the modified wave equation

five rows of these matrices to show how (7) continues for $i = 1, 2, 3, 4, 5$

$$\mathbf{G}_0^{\delta\Omega_{+1}} = \begin{bmatrix} 0 & G(1,17) & G(1,18) & G(1,19) & 0 & G(1,19) & G(1,20) & G(1,21) & 0 & G(1,21) & G(1,22) & G(1,23) & 0 & G(1,23) & G(1,24) & G(1,17) \\ 0 & G(2,17) & G(2,18) & G(2,19) & 0 & G(2,19) & G(2,20) & G(2,21) & 0 & G(2,21) & G(2,22) & G(2,23) & 0 & G(2,23) & G(2,24) & G(2,17) \\ 0 & G(3,17) & G(3,18) & G(3,19) & 0 & G(3,19) & G(3,20) & G(3,21) & 0 & G(3,21) & G(3,22) & G(3,23) & 0 & G(3,23) & G(3,24) & G(3,17) \\ 0 & G(4,17) & G(4,18) & G(4,19) & 0 & G(4,19) & G(4,20) & G(4,21) & 0 & G(4,21) & G(4,22) & G(4,23) & 0 & G(4,23) & G(4,24) & G(4,17) \\ 0 & G(5,17) & G(5,18) & G(5,19) & 0 & G(5,19) & G(5,20) & G(5,21) & 0 & G(5,21) & G(5,22) & G(5,23) & 0 & G(5,23) & G(5,24) & G(5,17) \end{bmatrix}, \quad (8)$$

$$\mathbf{G}_0^{\delta\Omega} = \begin{bmatrix} G(1,2)+G(1,16) & G(1,3) & G(1,4)+G(1,6) & G(1,7) & G(1,8)+G(1,10) & G(1,11) & G(1,12)+G(1,14) & G(1,15) \\ G(2,2)+G(2,16) & G(2,3) & G(2,4)+G(2,6) & G(2,7) & G(2,8)+G(2,10) & G(2,11) & G(2,12)+G(2,14) & G(2,15) \\ G(3,2)+G(3,16) & G(3,3) & G(3,4)+G(3,6) & G(3,7) & G(3,8)+G(3,10) & G(3,11) & G(3,12)+G(3,14) & G(3,15) \\ G(4,2)+G(4,16) & G(4,3) & G(4,4)+G(4,6) & G(4,7) & G(4,8)+G(4,10) & G(4,11) & G(4,12)+G(4,14) & G(4,15) \\ G(5,2)+G(5,16) & G(5,3) & G(5,4)+G(5,6) & G(5,7) & G(5,8)+G(5,10) & G(5,11) & G(5,12)+G(5,14) & G(5,15) \end{bmatrix}. \quad (9)$$

In Fig. 14 b, we see that the corner nodes on Ω_{+1} are involved twice while the corner nodes on Ω are never evaluated. This explains the zero columns in Eq. 8 and the columns with sums in Eq. 9.

As mentioned before, once we form these four sets of Green's function matrices, we solve (3), where we compute the forward wavefield in the following three steps: *(i)* Evaluate the vector of unknowns by solving equation 3, *(ii)* extract the scattered field on the boundary of the local domain and one layer to the interior, *(iii)* project the scattered wavefield $\delta\mathbf{u}$ on the receiver locations in the seismic acquisition using Eq. 2, *(iv)* add the precomputed background Green's function matrix $\mathbf{u}_0^{(s,r)}$ and the projected scattered wavefield to compute the forward wavefield at the receiver locations. We thus compute the data residual and the objective function for the locally perturbed model using the local solver exclusively. To compute the adjoint wavefield, we now propagate the residual back to the truncated domain. We do this by using the same sets of Green's functions that we used to project the scattered wavefield to the surface of the full domain. This is simply achieved by multiplying the Green's function matrix $\mathbf{u}_0^{\delta\Omega}$ with the data residual. We again solve (3) using the modified source function on the boundary of the truncated domain to compute the adjoint wavefield. Just like the forward wavefield, this adjoint wavefield is numerically exactly the same as would have been generated by a full domain solver on the perturbed model. Finally, we use the numerically exact forward and adjoint wavefields to compute the numerically exact gradients in the truncated domain, which is exactly the same FWI gradient as a full domain Helmholtz solver would have returned.

Appendix 2

Here, we illustrate a computationally efficient rank-minimization–based framework to solve the source-separation problem in Section 5.3.1, which is an important step to make the numerically exact local solver feasible for the large-scale 3D seismic data acquisition. Rank-minimization–based formulations are based upon the following three fundamental principles: *(i)* the underlying target matrix should exhibit low-rank structure in some transform domain, *(ii)* the subsampling-blending operator should increase the rank or slow down the decay of the singular values in some transform domain, *(iii)* a scalable and computationally efficient rank-minimization framework to handle large-scale data matrices. Under assumptions *(i)* and *(ii)*, the goal of the rank-minimization problem is to find the matrix of lowest possible rank that agrees with the experimental observations. This is written as the following optimization problem:

$$\underset{\mathbf{X}}{\text{minimize}} \quad \text{rank}(\mathbf{X}) \quad \text{subject to} \quad \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2 \leq \epsilon,$$

where rank is defined as the maximum number of linearly independent rows or column of a matrix, $\mathbf{b}$ is a set of blended measurements, and $\mathcal{A}$ represents the sampling-blending operator. Since rank-minimization problems are NP hard and therefore computationally intractable, [46] showed that solutions to rank-minimization problems can be found by solving the following nuclear norm minimization problem:

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{X}\|_* \quad \text{subject to} \quad \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2 \leq \epsilon, \qquad (10)$$

where $\|.\|_* = \|\boldsymbol{\sigma}\|_1$ and $\boldsymbol{\sigma}$ is the vector of singular values for each monochromatic data matricization. To efficiently solve equation 10 for large-scale seismic data, we used an extension of SPGℓ_1 solver [4] developed for basis-pursuit denoising (BPDN$_\sigma$) in [2]. The resulting algorithm which is dubbed SPG-LR by [2] finds the solution to the BPDN$_\sigma$ by solving a sequence of robust LASSO (least absolute shrinkage and selection operator) subproblems:

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2 \quad \text{subject to} \quad \|\mathbf{X}\|_* \leq \tau. \qquad (11)$$

where τ is updated by traversing the Pareto curve. The LASSO is a regularized regression formulation that seeks to do variable selection using a sparsity penalty, whereas the Pareto curve describes the tradeoff between the data fit and the nuclear norm of the solution vector. Interested readers can find a detailed explanation of the Pareto curve in [4]. Solving each robust LASSO subproblem requires a projection onto the nuclear norm ball $\|X\|_* \leq \tau$ in every iteration by performing a singular value decomposition and then thresholding the singular values. In the case of large-scale seismic problems, it becomes prohibitive to carry out such a large number of SVDs. Therefore, we avoid the direct approach to the nuclear norm minimization problem and follow a factorization-based approach [30, 47, 48]. The factorization-based approach parametrizes each monochromatic data matrix $\mathbf{X}$ as a product of two low-rank factors $\mathbf{L} \in \mathbb{C}^{n\times k}$ and $\mathbf{R} \in \mathbb{C}^{m\times k}$ such that $\mathbf{X} = \mathbf{L}\mathbf{R}^H$, where k represents the rank of the underlying matrix and ^{H} represents the Hermitian transpose. The optimization scheme can then be carried out using the matrices $\mathbf{L}, \mathbf{R}$ instead of $\mathbf{X}$, thereby significantly reducing the size of the decision variable from $n\times m$ to $k\times(n+m)$ when $k \leq (n, m)$. Rennie and Srebro [48] show that the nuclear norm obeys the relationship

$$\|\mathbf{X}\|_* \leq \frac{1}{2}\|\mathbf{L};\mathbf{R}\|_F^2,$$

where $\|\cdot\|_F^2$ is the Frobenius norm of the matrix (sum of the squared entries). Consequently, the LASSO subproblem can be replaced by

$$\min_{\mathbf{L},\mathbf{R}} \quad \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2 \quad \text{s.t.} \quad \frac{1}{2}\|\mathbf{L};\mathbf{R}\|_F^2 \leq \tau \ .$$

where the projection onto $\frac{1}{2}\|\mathbf{L};\mathbf{R}\|_F^2 \leq \tau$ is easily achieved by multiplying each factor $\mathbf{L}$ and $\mathbf{R}$ by the scalar $2\tau/(\frac{1}{2}\|\mathbf{L};\mathbf{R}\|_F^2)$.

References

1. Abma, R., Ford, A., Rose-Innes, N., Mannaerts-Drew, H., Kommedal, J.: Continued development of simultaneous source acquisition for ocean bottom surveys. In: 75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013 (2013)
2. Aravkin, A.Y., Kumar, R., Mansour, H., Recht, B., Herrmann, F.J.: Fast methods for denoising matrix completion formulations, with applications to robust seismic data interpolation. SIAM J. Sci. Comput. **36**(5), S237–S266 (2014)
3. Beasley, C.J.: A new look at marine simultaneous sources. Lead. Edge **27**(7), 914–917 (2008)
4. Berg, E.v., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**(2), 890–912 (2008)
5. Berryhill, J.R.: Wave-equation datuming before stack. In: SEG Technical Program Expanded Abstracts 1984, pp. 397–399. Society of Exploration Geophysicists (1984)
6. Broggini, F., Snieder, R., Wapenaar, K.: Data-driven wavefield focusing and imaging with multidimensional deconvolution: numerical examples for reflection data with internal multiples. Geophysics **79**(3), WA107–WA115 (2014)
7. Brougois, A., Bourget, M., Lailly, P., Poulet, M., Ricarte, P., Versteeg, R.: Marmousi, model and data. In: EAEG Workshop-Practical Aspects of Seismic Data Inversion (1990)
8. Byun, J., Yu, J., Seol, S.J.: Crosswell monitoring using virtual sources and horizontal wells. Geophysics **75**(3), SA37–SA43 (2010)
9. Candes, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
10. Da Silva, C., Herrmann, F.J.: A unified 2D/3D large scale software environment for nonlinear inverse problems. Tech. rep. (2017). Submitted to ACM Transactions on Mathematical Software on February, 14 (2017)
11. Denli, H., Huang, L.: Double-difference elastic waveform tomography in the time domain. In: SEG Technical Program Expanded Abstracts 2009, pp. 2302–2306. Society of Exploration Geophysicists (2009)
12. Dong, S., Luo, Y., Xiao, X., Chávez-Pérez, S., Schuster, G.T.: Fast 3D target-oriented reverse-time datuming. Geophysics **74**(6), WCA141–WCA151 (2009)
13. Donoho, D.L.: Compressed sensing. IEEE Trans. Inform. Theory **52**(4), 1289–1306 (2006)
14. Etgen, J.T., Chu, C., Yang, T., Vyas, M.: Adaptive image focusing. In: SEG Technical Program Expanded Abstracts 2014, pp. 3774–3778. Society of Exploration Geophysicists (2014)
15. Fokkema, J.T., van den Berg, P.M.: Seismic applications of acoustic reciprocity. Elsevier (1993)
16. Haber, E., Chung, M., Herrmann, F.: An effective method for parameter estimation with PDE constraints with multiple right-hand sides. SIAM J. Optim. **22**(3), 739–757 (2012)
17. Haffinger, P.R.: Seismic broadband full waveform inversion by shot/receiver refocusing (2013)
18. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011)
19. Herrmann, F.J., Friedlander, M.P., Yilmaz, O.: Fighting the curse of dimensionality: compressive sensing in exploration seismology. IEEE Signal Process. Mag. **29**(3), 88–100 (2012)
20. Johnston, D.H.: Practical applications of time-lapse seismic data. Society of Exploration Geophysicists (2013)
21. Jumah, B., Herrmann, F.J.: Dimensionality-reduced estimation of primaries by sparse inversion. Geophys. Prospect. **62**(5), 972–993 (2014)
22. Kadu, A., van Leeuwen, T., Mulder, W.A.: Salt reconstruction in full-waveform inversion with a parametric level-set method. IEEE Trans. Comput. Imag. **3**(2), 305–315 (2017)
23. Kreimer, N., Stanton, A., Sacchi, M.D.: Tensor completion based on nuclear norm minimization for 5D seismic data reconstruction. Geophysics **78**(6), V273–V284 (2013)
24. Kumar, R., Graff-Kray, M., Vasconcelos, I., Herrmann, F.J.: Target-oriented imaging using extended image volumes—a low-rank factorization approach. Geophysical Prospecting
25. Kumar, R., Silva, C.D., Akalin, O., Aravkin, A.Y., Mansour, H., Recht, B., Herrmann, F.J.: Efficient matrix completion for seismic data reconstruction. Geophysics **80**(05), V97–V114 (2015)
26. Kumar, R., Wason, H., Herrmann, F.J.: Source separation for simultaneous towed-streamer marine acquisition –- a compressed sensing approach. Geophysics **80**(06), WD73–WD88 (2015)
27. Kumar, R., Wason, H., Sharan, S., Herrmann, F.J.: Highly repeatable 3D compressive full-azimuth towed-streamer time-lapse acquisition –- a numerical feasibility study at scale. Lead. Edge **36**(8), 677–687 (2017)
28. Landrø, M.: Discrimination between pressure and fluid saturation changes from time-lapse seismic data. Geophysics **66**(3), 836–844 (2001)

29. Lange, M., Kukreja, N., Luporini, F., Louboutin, M., Yount, C., Hückelheim, J., Gorman, G.: Optimised finite difference computation from symbolic equations. In: Python in Science Conference Proceedings, p. 89–96 (2017)
30. Lee, J., Recht, B., Salakhutdinov, R., Srebro, N., Tropp, J.: Practical large-scale optimization for max-norm regularization. In: Advances in Neural Information Processing Systems, 2010 (2010)
31. van Leeuwen, T., Herrmann, F.J.: Fast waveform inversion without source-encoding. Geophys. Prospect. **61**(s1), 10–19 (2013)
32. van Leeuwen, T., Herrmann, F.J.: 3D frequency-domain seismic inversion with controlled sloppiness. SIAM J. Sci. Comput. **36**(5), S192–S217 (2014)
33. Lewis, W., Starr, B., Vigh, D.: A level set approach to salt geometry inversion in full-waveform inversion. In: SEG Technical Program Expanded Abstracts 2012, pp. 1–5. Society of Exploration Geophysicists (2012)
34. Li, X., Aravkin, A.Y., van Leeuwen, T., Herrmann, F.J.: Fast randomized full-waveform inversion with compressive sensing. Geophysics **77**(3), A13–A17 (2012)
35. Lumley, D.E.: Time-lapse seismic reservoir monitoring. Geophysics **66**(1), 50–53 (2001)
36. Malcolm, A., Willemsen, B.: Rapid 4D FWI using a local wave solver. Lead. Edge **35**(12), 1053–1059 (2016)
37. van Manen, D.J., Robertsson, J.O., Curtis, A.: Exact wave field simulation for finite-volume scattering problems. J. Acous. Soc. Am. **122**(4), EL115–EL121 (2007)
38. Masson, Y., Cupillard, P., Capdeville, Y., Romanowicz, B.: On the numerical implementation of time-reversal mirrors for tomographic imaging. Geophys. J. Int. **196**(3), 1580–1599 (2013)
39. Masson, Y., Romanowicz, B.: Fast computation of synthetic seismograms within a medium containing remote localized perturbations: a numerical solution to the scattering problem. Geophys. J. Int. **208**(2), 674–692 (2016)
40. Masson, Y., Romanowicz, B.: Box tomography: localized imaging of remote targets buried in an unknown medium, a step forward for understanding key structures in the deep earth. Geophys. J. Int. **211**(1), 141–163 (2017)
41. Mosher, C., Li, C., Morley, L., Ji, Y., Janiszewski, F., Olson, R., Brewer, J.: Increasing the efficiency of seismic data acquisition via compressive sensing. Lead. Edge **33**(4), 386–391 (2014)
42. Mulder, W.A.: Rigorous redatuming. Geophys. J. Int. **161**(2), 401–415 (2005)
43. van der Neut, J., Herrmann, F.J.: Interferometric redatuming by sparse inversion. Geophys. J. Int. **192**(2), 666–670 (2012)
44. Oosterlee, C., Vuik, C., Mulder, W., Plessix, R.E.: Shifted-Laplacian preconditioners for heterogeneous Helmholtz problems. In: Advanced Computational Methods in Science and Engineering, pp. 21–46. Springer (2009)
45. Plessix, R.E.: A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. Geophys. J. Int. **167**(2), 495–503 (2006)
46. Recht, B., Fazel, M., Parrilo, P.: Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
47. Recht, B., Ré, C.: Parallel stochastic gradient algorithms for large-scale matrix completion. In: Optimization Online (2011)
48. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 713–719. ACM, New York (2005)
49. Robertsson, J.O., Amundsen, L., Pedersen, Å.S.: Signal apparition for simultaneous source wavefield separation. Geophys. J. Int. **206**(2), 1301–1305 (2016)
50. Robertsson, J.O., Chapman, C.H.: An efficient method for calculating finite-difference seismograms after model alterations. Geophysics **65**(3), 907–918 (2000)
51. Schmidt, M., Berg, E., Friedlander, M., Murphy, K.: Optimizing costly functions with simple constraints: a limited-memory projected quasi-newton algorithm. In: Artificial Intelligence and Statistics, pp. 456–463 (2009)
52. Schuster, G.T., Zhou, M.: A theoretical overview of model-based and correlation-based redatuming methods. Geophysics **71**(4), SI103–SI110 (2006)
53. Silva, C.D., Herrmann, F.J.: Optimization on the hierarchical Tucker manifold - applications to tensor completion. Linear Algebra Appl. **481**, 131–173 (2015). https://doi.org/10.1016/j.laa.2015.04.015. (Linear Algebra and its Applications)
54. Tang, Y., Biondi, B.: Target-oriented wavefield tomography using synthesized born data. Geophysics **76**(5), WB191–WB207 (2011)
55. Vermeer, G.J.: 3-D seismic survey design. Society of Exploration Geophysicists (2002)
56. Virieux, J., Operto, S.: An overview of full-waveform inversion in exploration geophysics. Geophysics **74**(6), WCC1–WCC26 (2009)
57. Wason, H., Oghenekohwo, F., Herrmann, F.J.: Low-cost time-lapse seismic with distributed compressive sensing—part 2: impact on repeatability. Geophysics **82**(3), P15–P30 (2017)
58. Willemsen, B., Malcolm, A., Lewis, W.: A numerically exact local solver applied to salt boundary inversion in seismic full-waveform inversion. Geophys. J. Int. **204**(3), 1703–1720 (2016)
59. Yang, D., Liu, F., Morton, S., Malcolm, A., Fehler, M.: Time-lapse full-waveform inversion with ocean-bottom-cable data: application on Valhall field. Geophysics **81**(4), R225–R235 (2016)
60. Yuan, S., Fuji, N., Singh, S., Borisov, D.: Localized time-lapse elastic waveform inversion using wavefield injection and extrapolation: 2-D parametric studies. Geophys. J. Int. **209**(3), 1699–1717 (2017)
61. Zhang, Y., Silva, C.D., Kumar, R., Herrmann, F.J.: Massive 3D seismic data compression and inversion with hierarchical Tucker. In: SEG Technical Program Expanded Abstracts, pp. 1347–1352 (2017)