SIAM J. SCI. COMPUT.
Vol. 36, No. 5, pp. S192–S217

3D FREQUENCY-DOMAIN SEISMIC INVERSION WITH CONTROLLED SLOPPINESS*

TRISTAN VAN LEEUWEN† AND FELIX J. HERRMANN‡

Abstract. Seismic waveform inversion aims at obtaining detailed estimates of subsurface medium parameters, such as the spatial distribution of soundspeed, from multiexperiment seismic data. A formulation of this inverse problem in the frequency domain leads to an optimization problem constrained by a Helmholtz equation with many right-hand sides. Application of this technique to industry-scale problems faces several challenges: First, we need to solve the Helmholtz equation for high wave numbers over large computational domains. Second, the data consist of many independent experiments, leading to a large number of PDE solves. This results in high computational complexity both in terms of memory and CPU time as well as input/output costs. Finally, the inverse problem is highly nonlinear and a lot of art goes into preprocessing and regularization. Ideally, an inversion needs to be run several times with different initial guesses and/or tuning parameters. In this paper, we discuss the requirements of the various components (PDE solver, optimization method, ...) when applied to large-scale three-dimensional seismic waveform inversion and combine several existing approaches into a flexible inversion scheme for seismic waveform inversion. The scheme is based on the idea that in the early stages of the inversion we do not need all the data or very accurate PDE solves. We base our method on an existing preconditioned Krylov solver (CARP-CG) and use ideas from stochastic optimization to formulate a gradient-based (quasi-Newton) optimization algorithm that works with small subsets of the right-hand sides and uses inexact PDE solves for the gradient calculations. We propose novel heuristics to adaptively control both the accuracy and the number of right-hand sides. We illustrate the algorithms on synthetic benchmark models for which significant computational gains can be made without being sensitive to noise and without losing the accuracy of the inverted model.

Key words. seismic inversion, Helmholtz equation, preconditioning, Kaczmarz method, inexact gradient, block-cg

AMS subject classifications. 65N21, 65K10

DOI. 10.1137/130918629

1. Introduction.

Detailed estimates of subsurface properties such as soundspeed and density can be obtained from seismic data by solving a PDE-constrained optimization problem [44]—also known as full-waveform inversion (FWI) in the seismic community—that involves multiple source experiments. The data in this setting consist of a collection of time series for many source-receiver pairs and are the result of either passive or active source experiments. The PDE is a wave equation with as many right-hand sides as there are sources. Applications of this technique include oil and gas exploration and global seismology. A common characteristic of these applications is the need to propagate the waves over large distances (i.e., several hundred wavelenghts) through strongly inhomogeneous media for a large number of sources.

*Received by the editors April 26, 2013; accepted for publication (in revised form) March 14, 2014; published electronically October 30, 2014. This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery grant (22R81254) and the Collaborative Research and Development grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, Chevron, CGG, ConocoPhillips, ION, Petrobras, PGS, Total SA, WesternGeco, and Woodside.
http://www.siam.org/journals/sisc/36-5/91862.html

†Department of Earth, Ocean and Atmospheric Sciences, University of British Columbia, Vancouver, BC, Canada. Current address: Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands (t.van.leeuwen@cwi.nl).

‡Department of Earth, Ocean and Atmospheric Sciences, University of British Columbia, Vancouver, BC, Canada (fherrmann@eos.ubc.ca).

This leads to a nonlinear data-fitting problem involving millions, and in industrial settings even billions, of unknowns and exceedingly large data volumes (up to $\sim 10^{15}$ data points).

While the FWI problem is naturally formulated in the time domain, we can formulate an equivalent problem in the frequency domain by applying a temporal Fourier transform to the observed data and using the Helmholtz equation to predict the data. One of the main reasons to do this is that it allows us to work with relatively small data volumes (i.e., a few frequencies instead of the whole time series). This formulation also avoids the need for input/output (i/o)-intensive checkpointing schemes when employing adjoint-state methods [42]. While iteratively solving the Helmholtz equation in three dimensions is still a challenge, recent benchmark tests suggest that it is becoming competitive when compared to time-stepping methods [25].

Although conceptually attractive, this straightforward data-fitting approach to seismic inversion is plagued by the severe nonlinear relation between the data and unknown medium parameters. In particular, the oscillatory nature of the data may result in a partial fit even though the medium parameters do not represent the true earth. This phenomenon is known as *loop skipping* and may cause gradient-based optimization methods to fail because they get stuck in a local minimum (see [43] for an overview). Many alternative formulations of the inverse problem have been proposed to mitigate this problem [51]. These formulations rely either on using different objective functions to measure the data misfit [27, 52] or on completely different formulations of the inverse problem [41, 34, 50]. While less sophisticated than the aforementioned methods, a simple remedy is based on the observation that loop skipping is less of a problem at low frequencies. This observation motivated [6, 37, 53] to propose a multiscale continuation method where the inversion is carried out from low to high frequencies. Needless to say, this multiscale approach is trivially implemented in the frequency domain.

PDE-constrained optimization problems can be solved in a variety of ways, via either Lagrange methods or SQP [19]. All these methods rely—in one way or another—on simultaneous updates of both the state and control variables, and thus avoids having to explicitly solve the PDE. Unfortunately, the scale of typical seismic problems, even in two dimensions, precludes the use of such *full-space* methods because it requires simultaneous storage of the forward and adjoint wavefields for all sources. For that reason, practitioners usually resort to *reduced-space* methods that rely on explicitly eliminating the state variables [36, 10] by solving the PDE for each gradient update. The gradient of the reduced objective with respect to the medium parameters can then be efficiently computed via the adjoint-state method [44, 33].

For two-dimensional (2D) applications, the discretized PDE (the Helmholtz equation) is usually factorized and its inverse can thus be pplied cheaply to multiple right-hand sides. In three dimensions, however, direct factorization does not scale very well to large problems due to the memory requirements caused by the large number of grid points needed and the increased bandwidth of the matrix. This poses major problems when applying three-dimensional (3D) waveform inversion to industry-scale datasets. (i) The Helmholtz equation is notoriously difficult to solve due to the indefiniteness and requires sophisticated preconditioners (see [12] for an extensive overview). This is a very active area of research and many preconditioners have been proposed. However, many of these approaches are problem specific, require (manual) tuning of problem-specific parameters to ensure convergence, and may suffer from significant setup and memory storage costs. While well suited for very accurate simulation with fixed medium parameters, these dedicated approaches are not particularly attractive

for inversion where the coefficients in the PDE change after every model update. (ii) The computational costs scale linearly with the number of sources, which in realistic settings makes even a single gradient computation prohibitively expensive. Recently, techniques from *stochastic optimization* have been applied to dramatically reduce the computational cost by lowering the number of right-hand sides via random projections or random subsampling [20, 46, 49, 55]. Aside from decreasing the computational cost by reducing the required number of PDE solves, these approaches also decrease the computational overhead related to i/o. Although this approach can lead to significant reduction of the per-iteration cost of gradient-based optimization algorithms, these stochastic methods have the disadvantage that they often lead to a sublinear convergence and hence may require disproportionally more iterations to converge. In theory, this loss of convergence can be overcome by gradually increasing the sample size [13] by bringing in more right-hand sides. The rate with which the sample size needs to be increased depends on problem-specific constants that are expensive to compute in practice. Hence, there is a need for an adaptive strategy based on heuristics that is easy to compute. Inexactness is another powerful tool for reducing the computational costs when using iterative methods to solve the PDEs [23, 4, 22]. It is not clear how to choose the tolerance of the inexact PDE solves such that the computations are still sufficiently accurate to be useful.

To carry out our data-intensive large-scale inversions based on local derivatives, iterative 3D PDE solves for time-harmonic Helmholtz, and stochastic optimization, we propose that a scalable—both in terms of data and model size—inversion strategy needs to consist of the following key ingredients:

- an iterative *Helmholtz solver* with low memory imprint and computational overhead, e.g., setup costs, whose convergence does not critically depend on model-dependent tuning parameters;
- a practical *stopping criterion* for the iterative solver that avoids computing accurate solutions when they are not needed—e.g., when the model iterate is far from the true solution;
- a (stochastic) *optimization strategy* that exploits the separable structure of the objective by working with small subsets of the right-hand sides at each iteration;
- a criterion to adaptively grow the *sample size* as the optimization proceeds; and
- the ability to exploit both model space parallelism via *domain decomposition* and *data-space parallelism*, by parallelizing loops over frequencies and/or right-hand sides.

Because practical application of FWI to field data still requires a lot of parameter tuning and careful selection of the initial model, the ultimate goal is to reduce the turnaround time for a typical inversion. This would allow us to run multiple scenarios to test the validity of the output.

1.1. Contributions. In this paper, we propose a new comprehensive framework for 3D seismic waveform inversion that addresses both the need to efficiently solve the Helmholtz equation as well as the computational cost induced by the many right-hand sides. To this end, we combine several existing components into a novel algorithm that addresses these issues by developing practical algorithms designed to scale well by exploiting data as well as model-space parallelism.

Helmholtz solver. Because of its relative simplicity, flexibility towards different types of wave physics, and robustness with respect to tuning parameter selection, we

investigate the use of an existing Kaczmarz-based preconditioned iterative method (CGMN) and more specifically its parallel extensions (CARP-CG). We evaluate the performance of this solver in the context of inversion. Because we are dealing with multiple right-hand sides, we incorporate the preconditioner in a block iterative method and evaluate its performance.

Inexact PDE solves. To reduce the computational cost of the inversion, we consider an *inexact* approach which solves the PDEs up to some relatively high prescribed tolerance. We propose a novel heuristic to adaptively determine the required accuracy of the PDE solves based on the data misfit.

Source subsampling and stochastic optimization. We further reduce the computational cost by working with small subsets of the sources at each iteration and propose a novel heuristic to determine whether the sample size needs to be increased based on an average descent of the (sampled) objective.

The result is a new adaptive stochastic quasi-Newton optimization method for seismic inversion that automatically determines the required accuracy for the PDE solves and automatically increases the sample size when an average descent condition fails. Since accuracy is really a proxy for computational costs, the basic motto of our approach is to never use more accuracy than strictly needed at any point during the inversion. This leads to significant reductions in the number of right-hand sides for the gradient calculations and in the number of iterations for the Helmholtz solves.

1.2. Outline. In section 2, we discuss the discretization of the Helmholtz equation and the Kaczmarz-based preconditioner. Section 3 describes the stochastic inversion strategy and the heuristics used to determined the accuracy of the PDE solves and the sample size. Some numerical experiments are presented in section 4 and section 5 presents conclusions and discussions.

2. Modeling. We model wave propagation in the earth via the following 3D scalar Helmholtz equation,

$$\left[\frac{\omega^2}{c(x)^2\rho(x)} + \nabla \cdot \frac{1}{\rho(x)}\nabla\right] u(\omega, x) = s(\omega, x) \ + \ \text{b.c.}'\text{s}, \tag{2.1}$$

where ω is the frequency, x is the spatial coordinate, $c(x)$ denotes the sound speed, $\rho(x)$ denotes the density, $u(\omega, x)$ denotes the wavefield, and $s(\omega, x)$ is a source term.

We discretize the Helmholtz equation using a a 27-point mixed-grid stencil with perfectly matched layer (PML) boundary conditions on all sides of the domain [31] and denote the resulting sparse linear system by

$$A\mathbf{u} = \mathbf{s}, \tag{2.2}$$

where $\mathbf{u}$ and $\mathbf{s}$ denote the gridded wavefield and source function.

2.1. Iterative solver. Due to the sheer size of typical problems in 3D seismic imaging ($\sim 10^9$ grid points), direct factorization techniques are not widely used. Recent work by Wang, de Hoop, and Xia [54] aims at reducing the memory imprint by employing hierarchical low-rank (HSS) representations of the factorization. Even taking these new developments into account, direct factorization techniques are not practical for the proposed inversion scheme since the initial factorization costs cannot be amortized over a large number of sources (we will focus on the use of optimization methods that use only a relatively small number of right-hand sides at each iteration). Moreover, the massively parallel implementation required for such direct methods

precludes the use of course-scale parallelism (e.g., parallelizing over sources and/or frequencies). Moreover, a new factorization will be needed after each model update.

Iterative techniques are much more efficient in terms of memory use, and are more attractive when using only a relatively small number of right-hand sides. However, they require a good preconditioner. Many preconditioning techniques have been proposed in the literature [11, 38, 32, 21, 9, 40, 35]. Most of these are designed for efficient and accurate forward modeling and include problem-specific tuning parameters to ensure convergence. The latter does not make these approaches very attractive for inversion since the medium parameters will change from one iteration to the next, possibly requiring the tuning parameters to be changed as well. Instead, we use CARP-CG [14, 15], a generic, iterative solution technique for sparse linear systems based on the CGMN method [5]. Convergence of this method is guaranteed, making it an attractive solver for inversion purposes. Moreover, the method is generic and can thus be applied to other discretizations or even vector Helmholtz equations without any modification.

Naively, one can think of this method as a Kaczmarz-preconditioned conjugate gradient (CG) method. For the sake of completeness, we give a brief overview of the method.

CGMN. The Kaczmarz method solves a system of N equations, $A\mathbf{u} = \mathbf{s}$, by cyclically projecting the iterate onto rows of the matrix [24]

$$\mathbf{u} := \mathbf{u} + \gamma \left(s_i - \mathbf{a}_i{}^* \mathbf{u}\right) \mathbf{a}_i / ||\mathbf{a}_i||_2^2, \quad i = 1 \dots N, \tag{2.3}$$

where $\mathbf{a}_i$ denotes the ith row of A as column vector, $\cdot^*$ denotes the conjugate transpose, and $0 < \gamma < 2$ is a relaxation parameter. Introducing the matrices $Q_i = \left(I - \gamma \mathbf{a}_i \mathbf{a}_i{}^* / ||\mathbf{a}_i||_2^2\right)$, we may write this iteration as

$$\mathbf{u} := Q_i \mathbf{u} + \gamma s_i \mathbf{a}_i / ||\mathbf{a}_i||_2^2.$$

A *double sweep* through the matrix (from row 1 to N and back) can then be denoted by

$$\mathbf{u} := Q\mathbf{u} + R\mathbf{s},$$

where $Q = Q_1 Q_2 \dots Q_N Q_N \dots Q_1$ and R contains all the factors multiplying $\mathbf{s}$. It is easily verified that the Q_i are Hermitian rank 1 matrices with eigenvalue $1 - \gamma$. It follows that Q is Hermitian and has eigenvalues $\in (-1, 1)$.

We may now transform the original system of equations to a Hermitian positive semidefinite system

$$(I - Q)\mathbf{u} = R\mathbf{s},$$

which we can solve with the CG method.

Since this preconditioner is equivalent to SSOR on the normal equations AA^* [5], we find the following alternative expressions for Q and R:

$$Q = I - A^* H A, \tag{2.4}$$

$$R = A^* H, \tag{2.5}$$

with $H = \gamma(2 - \gamma) \left(D + \gamma L^*\right)^{-1} D \left(D + \gamma L\right)^{-1}$, where D and L contain the diagonal and lower triangular elements of AA^*. Although these expressions are convenient for analysis and testing, a more efficient implementation computes only the action of these matrices using Algorithm 1 as $Q\mathbf{u} + R\mathbf{s} = \mathsf{DKSWP}(A, \mathbf{u}, \mathbf{s}, \gamma)$ [14].

The choice of the parameter γ does not seem to critically affect the convergence, although values $\gamma \approx 1.5$ appear to be optimal (see, for example, [16] for extensive

ALGORITHM 1. DKSWP$(A, \mathbf{u}, \mathbf{s}, \gamma)$ PERFORMS A FORWARD AND BACKWARD KACZMARZ SWEEP ON THE MATRIX A.

{forward sweep}
for $i = 1$ to N **do**
  $\mathbf{u} := \mathbf{u} + \gamma(s_i - \mathbf{a}_i{}^*\mathbf{u})\mathbf{a}_i/||\mathbf{a}_i||_2^2$
end for
{backward sweep}
for $i = N$ to 1 **do**
  $\mathbf{u} := \mathbf{u} + \gamma(s_i - \mathbf{a}_i{}^*\mathbf{u})\mathbf{a}_i/||\mathbf{a}_i||_2^2$
end for
return u

numerical experiments). A Fourier analysis of the one-dimensional Helmholtz equation of the CGMN method confirms this [47].

Finally, the method lends itself to a matrix-free implementation where the stencil coefficients are generated on-the-fly. This will be important when moving to higher-order stencils where we cannot afford to store all the stencil coefficients.

Block-CG. Since we are mostly dealing with multiple right-hand sides simultaneously, we use a block-CG method [3, 18] to solve the Helmholtz equation for multiple right-hand sides simultaneously. By building up a Krylov subspace using multiple residual vectors simultaneously, block iterative methods may converge significantly faster at the cost of an increased computational cost per iteration. It is well known that block iterative methods will break down if the residual vectors become linearly dependent. To counter this, a reorthogonalization of the residuals in conjunction with so-called variable-block or deflation techniques can be used [3, 29, 8]. However, since the right-hand sides in our case represent different sources we expect the initial residuals to be nearly orthogonal. In our experiments we typically reach the desired tolerance—which is usually high—before any linear dependence between the residuals develops. The resulting algorithm—without deflation—is given in Algorithm 2 (adapted from [18, algorithm 8]). For a more robust implementation, variable-block or deflation techniques can be added.

ALGORITHM 2. BCGMN$(A, U_0, S, \gamma, \epsilon)$ BLOCK-CGMN ALGORITHM ON SYSTEM $AU = S$ USING DKSWP TO PERFORM THE MATRIX-VECTOR MULTIPLICATIONS.

$P_0 = R_0 = \mathsf{DKSWP}(A, U_0, S, \gamma) - U_0$
while $||R_k||_F > \epsilon||R_0||_F$ **do**
  $Q_k = P_k - \mathsf{DKSWP}(A, P_k, 0, \gamma)$
  $\alpha_k = ({P_k}^*Q_k)^{-1}({R_k}^*R_k)$
  $U_{k+1} = U_k + P_k\alpha_k$
  $R_{k+1} = R_k - Q_k\alpha_k$
  $\beta_k = ({R_k}^*R_k)^{-1}({R_{k+1}}^*R_{k+1})$
  $P_{k+1} = R_k + P_k\beta_k$
  $k = k + 1$
end while

Domain decomposition. The CARP-CG algorithm is a parallel extension of the CGMN algorithm, where the Kaczmarz projections are done independently and in parallel on blocks of rows. Between each sweep through the rows, the overlapping elements of the solutions are averaged. As such, its implementation resembles an additive Schwarz approach, however, CARP-CG is guaranteed to converge. For details

we refer the reader to [14, 15]. The use of CARP-CG to solve the Helmholtz equation specifically is described by [16, 17].

In summary, we discussed a generic solver for the Helmholtz equation with multiple right-hand sides. Because convergence of the method does not critically depend on tuning parameters, this method is suitable for inversion. Some numerical experiments and a comparison to other simple iterative methods are presented in section 4.1.

3. Inversion. The inverse problem for M sources and a single frequency can be cast as a PDE-constrained optimization problem

$$\min_{\mathbf{m},\mathbf{w},\mathbf{u}} \quad \sum_{i=1}^{M} \rho\big(w_i P_i \mathbf{u}_i - \mathbf{d}_i\big) \quad \text{s.t.} \quad A(\mathbf{m})\mathbf{u}_i = \mathbf{s}_i, \tag{3.1}$$

where ρ is a (differentiable) penalty function, $\mathbf{m}$ denotes the model parameter of interest (i.e, velocity, density, or some combination thereof), A is the discretized Helmholtz operator, $\mathbf{u} = [\mathbf{u}_1; \ldots; \mathbf{u}_M]$ are the wavefields for sources $\mathbf{s} = [\mathbf{s}_1; \ldots; \mathbf{s}_M]$, P_i is a detection operator, $\mathbf{w}$ is a vector of source weights, and $\mathbf{d} = [\mathbf{d}_1; \ldots; \mathbf{d}_M]$ are the observed data.

Standard *all-at-once* approaches to solving this PDE-constrained optimization problem rely on a Lagrangian formulation of this problem,

$$L(\mathbf{m}, \mathbf{w}, \mathbf{u}, \mathbf{v}) = \sum_{i=1}^{M} \rho\big(w_i P_i \mathbf{u}_i - \mathbf{d}_i\big) + \mathbf{v}_i^* \big(A(\mathbf{m})\mathbf{u}_i - \mathbf{s}_i\big), \tag{3.2}$$

and use a Newton-like method to solve $\nabla L = 0$, where

$$\nabla_{\mathbf{u}_i} L = P_i^* \nabla \rho\big(w_i P_i \mathbf{u}_i - \mathbf{d}_i\big) + A(\mathbf{m})^* \mathbf{v}_i, \tag{3.3}$$

$$\nabla_{\mathbf{v}_i} L = A(\mathbf{m})\mathbf{u}_i - \mathbf{s}_i, \tag{3.4}$$

$$\nabla_{w_i} L = \big(P_i \mathbf{u}_i\big)^* \nabla \rho\big(w_i P_i \mathbf{u}_i - \mathbf{d}_i\big), \tag{3.5}$$

$$\nabla_{\mathbf{m}} L = \sum_{i=1}^{M} G(\mathbf{m}, \mathbf{u}_i)^* \mathbf{v}_i, \tag{3.6}$$

where $G(\mathbf{m}, \mathbf{u}_i) = \frac{\partial A(\mathbf{m})\mathbf{u}_i}{\partial \mathbf{m}}$. Updating both state and control variables is not feasible for such large-scale problems so we use a so-called *reduced* approach, eliminating the state variables $\mathbf{u}_i$ and $\mathbf{v}_i$ by solving the forward ($\nabla_{\mathbf{v}_i} L = 0$) and adjoint ($\nabla_{\mathbf{u}_i} L = 0$) PDEs. We simplify the problem even further by projecting out $\mathbf{w}$. This can be done efficiently by solving a series of scalar optimization problems [2]

$$w_i = \operatorname*{argmin}_{w} \rho\big(w P_i \mathbf{u}_i - \mathbf{d}_i\big). \tag{3.7}$$

Finally, the reduced objective is given by

$$\phi(\mathbf{m}) = \sum_{i=1}^{M} \phi_i(\mathbf{m}), \quad \phi_i(\mathbf{m}) = \rho\big(w_i P_i \mathbf{u}_i - \mathbf{d}_i\big). \tag{3.8}$$

Note that the evaluation of the reduced objective requires the solution of the forward PDE (3.4) as well as the solution of M scalar optimization problems (3.7). The gradient of ϕ coincides with the gradient of the Lagrangian w.r.t. $\mathbf{m}$ (3.6) evaluated at the optimal $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{w}$ and thus requires the additional solution of an adjoint PDE (3.3).

3.1. Gradient descent with errors. A basic gradient-descent algorithm to minimize $\phi(\mathbf{m})$ is based on the iteration

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \nu_k \nabla\phi(\mathbf{m}_k), \tag{3.9}$$

where ν_k is the step size. However, the evaluation of the reduced objective (3.8) and its gradient requires 2M PDE solves, which may be prohibitively large. To reduce these costs, we resort to an optimization technique described by [13] that allows for the use of approximate gradients in a gradient descent algorithm:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \nu_k \mathbf{g}_k, \tag{3.10}$$

where $\mathbf{g}_k = \nabla\phi(\mathbf{m}_k) + \mathbf{e}_k$ is the approximate gradient and $\mathbf{e}_k$ is the approximation error. Specifically, [13, Thm. 2.2] proves that under some convexity assumptions on the objective function $\phi(\mathbf{m})$ and for an appropriately chosen fixed step size, a basic gradient-descent algorithm with approximate gradients (3.10) will converge as

$$\phi(\mathbf{m}_k) - \phi(\mathbf{m}_*) < a_k(\phi(\mathbf{m}_0) - \phi(\mathbf{m}_*)),$$

where $a_k = \max\{c^k, ||\mathbf{e}_k||_2^2\}$, where $0 \leq c < 1$ depends on the condition number of ϕ (c close to one being ill-conditioned). Thus, if the error in the gradient decreases linearly, the resulting gradient-descent algorithm with approximate gradients will also converge linearly. Furthermore, this tells us that we need not decrease the accuracy very fast if the problem is ill-conditioned to begin with (i.e., if c is close to one).

Of course, our objective is not very likely to satisfy these convexity assumptions *globally*, however, for the sake of designing an algorithm we will assume them to hold *locally*.

Source subsampling. We can obtain approximate gradients with controllable error by using only a subset of terms $\mathcal{I} \subseteq \{1, 2, \ldots, M\}$ in (3.8) when calculating the gradient. This *sample average* approach is extensively analyzed by [13, 1]. If we choose the elements in $\mathcal{I}$ randomly from $[1, 2, \ldots, M]$ (without replacement), the *expected* error in the gradient can be expressed as $||\mathbf{e}||_2 \propto \sqrt{b^{-1} - M^{-1}}$ [13, section 3.2]. The computational cost is directly proportional to the sample size $b = |\mathcal{I}|$, and thus a higher error in the gradient directly translates into a lower computational cost. Numerical experiments have shown that this approach is beneficial on 2D seismic inversion problems [49, 28]. How to choose the rate of increase of the sample size in practice is an open problem. Van den Doel and Ascher [46] suggest a cross-validation technique that relies on computing the objective for two independent samples and requiring a decrease on both. When this condition fails, the authors suggest doubling the sample size. This leads to an exponentially increasing sample size and induces a lot of computational overhead since all the computations are carried out twice.

We adopt a related approach that induces less computational overhead and increases the sample size at a slower (linear) rate; we redraw the samples at each iteration and use the misfit for both the old and new samples to keep track of the average descent. If the objective fails to decrease on average, we increase the sample size by a constant amount. Details of the implementation in the context of an L-BFGS method are described in section 3.3.

3.2. Approximate PDE solves. In addition to using source-subsampling techniques to reduce the computational cost, we can solve the forward and adjoint PDEs up to some (high) tolerance ϵ. The crux lies in determining a reasonable ϵ. The use of inexact PDE solves in the context of trust-region methods is discussed by [23, 22], while [4] discuss inexactness issues in full-space approaches.

Here we propose a heuristic to estimate a reasonable tolerance that is "accurate enough" and is completely independent of the optimization strategy and allows for the use of black-box optimization methods. The heuristic is based on the behavior of the objective for a single experiment as a function of the tolerance

$$\phi_i(\mathbf{m},\epsilon) = \rho\big(w_i P_i \mathbf{u}_i(\epsilon) - \mathbf{d}_i\big),$$

where $\mathbf{u}_i(\epsilon)$ is obtained by solving the forward PDE with tolerance ϵ (i.e., $||A\mathbf{u}_i(\epsilon) - \mathbf{s}_i||_2 \leq \epsilon||\mathbf{s}_i||_2$). We propose to choose a tolerance that computes the residual to within a certain fraction η of its true value, i.e., find an ϵ such that

$$|\phi_i(\mathbf{m},\epsilon) - \phi_i(\mathbf{m},0)| \leq \eta\phi_i(\mathbf{m},0).$$

This is not very practical since it would require a very accurate solve (with $\epsilon = 0$). Instead, we find a k such that

$$|\phi_i(\mathbf{m},\alpha^k\epsilon) - \phi_i(\mathbf{m},\alpha^{k+1}\epsilon)| \leq \eta\phi_i(\mathbf{m},\alpha^{k+1}\epsilon),$$

where $\alpha < 1$. The resulting algorithm to evaluate the objective for a given sample $\mathcal{I}$ is shown in Algorithm 3. We use the current solution, $\mathbf{u}_i(\alpha^k\epsilon)$, as an initial guess when solving for $\mathbf{u}_i(\alpha^{k+1}\epsilon)$ in line 9. We use the same tolerance when solving the adjoint PDE.

Of course, a similar heuristic can be used to estimate the actual error in the gradient by finding a k such that

$$||\mathbf{g}_i(\alpha^k\epsilon) - \mathbf{g}_i(\alpha^{k+1}\epsilon)||_2 \leq \eta||\mathbf{g}_i(\alpha^{k+1}\epsilon)||_2,$$

where $\mathbf{g}_i(\epsilon) = G(\mathbf{m},\mathbf{u}_i(\epsilon))^*\mathbf{v}_i(\epsilon)$ and $\mathbf{v}_i(\epsilon)$ is obtained by solving the adjoint PDE up to ϵ. Perhaps it would be even better to use two different tolerances for the forward and adjoint PDEs, but since it was seen that the heuristic based on the misfit yielded very good results, we will leave detailed comparison of these two heuristics for future research.

3.3. A stochastic quasi-Newton algorithm. We incorporate the heuristics described above in a novel stochastic quasi-Newton algorithm. Aside from using only a subset of the sources and approximate PDE solves, we redraw the sample at each iteration, even if the sample size does not increase. Such a renewal of the sample removes any possible bias introduced by using a particular subset and has proved highly beneficial in seismic applications [48, 26].

The algorithm decreases η when the line search fails and increases the sample size when the *average* objective as computed over two independent samples does not decrease. A detailed description is given in Algorithm 4.

The objective values f'_k and f'_{k+1} in line 29 are computed using different samples because the samples themselves are renewed at each iteration (cf. line 23), even if the sample size remains unchanged.

For the line search (line 15) we use a weak Wolfe line search and lbfgs (line 12) applies the L-BFGS scaling based on past gradient and model updates (cf. lines 19,20) [30]. Note that the Hessian update requires an extra gradient computation since we want to compute the update vectors $\mathbf{y}_k$ (line 20) from gradients that were computed for the same sample $\mathcal{I}$ [39, 7].

The above described adaptive stochastic algorithm combines several features needed to solve large-scale seismic inversion problems. These include (i) working with small subsets of data to limit the number of PDE solves and i/o, (ii) control over the accuracy of the PDE solves to reduce the number of CG iterations,

ALGORITHM 3 $\{f, \mathbf{g}\} = \mathsf{misfit}(\mathbf{m}, \mathcal{I}, \eta)$. COMPUTES THE MISFIT AND GRADIENT WITH AN AUTOMATICALLY CHOSEN TOLERANCE FOR THE PDE SOLVES.

{Initialization}
$f = 0$, $\mathbf{g} = 0$
$\epsilon = 10^{-2}$
$\alpha = 0.5$

{loop over sources (in parallel)}
for $i \in \mathcal{I}$ **do**
  for $k = 0 \to 10$ **do**
    {solve forward equation}
    solve $A(\mathbf{m})\mathbf{u}_i = \mathbf{s}_i$ using CGMN with tolerance ϵ

    {solve for source-weight}
    $w = \text{argmin}_w \, \rho(wP_i\mathbf{u}_i - \mathbf{d}_i)$

    {compute residual}
    $r_k = \rho(wP_i\mathbf{u}_i - \mathbf{d}_i)$

    {check residual}
    if $|r_k - r_{k-1}| \leq \eta r_k$ **then**
      break
    else
      $\epsilon = \alpha\epsilon$
    end if
  end for

  {solve adjoint equation}
  solve $A(\mathbf{m})^*\mathbf{v}_i = P_i{}^*\nabla\rho(wP_i\mathbf{u}_i - \mathbf{d}_i)$ using CGMN with tolerance ϵ

  {compute misfit and gradient}
  $f = f + |\mathcal{I}|^{-1}\rho(wP_i\mathbf{u}_i - \mathbf{d}_i)$
  $\mathbf{g} = \mathbf{g} + |\mathcal{I}|^{-1}G(\mathbf{m}, \mathbf{u}_i)^*\mathbf{v}_i$
end for

(iii) control over the error in the gradients by adaptively increasing the sample size when an average descent condition fails.

4. Numerical results. The algorithms are implemented in (parallel) MATLAB, with the exception of the DKSWP function, which is implemented in C and called from MATLAB using a MEX interface. The implementation of DKSWP makes use of a band-storage format of the matrix, thus allowing for efficient access to the rows of the matrix while minimizing storage overhead associated with generic sparse matrix storage formats.

4.1. CARP-CG. We illustrate the properties of the iterative method numerically on a well-known seismic benchmark model (the Overthrust model) depicted in Figure 1(a). The corresponding wavefield at 4 Hz for a point source is shown in Figure 1(b). For all the experiments, we adapt the grid spacing to the frequency to ensure a minimum of 10 grid points per wavelength.

CGMN versus other iterative methods. We compare CGMN to BiCGstab and restarted GMRES, both with an ILU(0) preconditioner. While other preconditioners are definitely more suitable for Helmholtz equations, our aim is to compare CGMN with methods of similar simplicity.

ALGORITHM 4. STOCHASTIC L-BFGS METHOD.

{Initialize}
$\eta = 0.1$
$b = 1$, $\beta = 1$, $b_{\max} = M$
choose $\mathcal{I}_0 \subseteq \{1, 2, \ldots, M\}$ s.t. $|\mathcal{I}_0| = b$

{evaluate misfit and gradient at initial guess}
$\{f_0, \mathbf{g}_0\} = \mathsf{misfit}(\mathbf{m}_0, \mathcal{I}_0, \eta)$

while not converged **do**

  {apply L-BFGS Hessian with history size m}
  $\delta\mathbf{m}_k = \mathsf{lbfgs}(-\mathbf{g}_k, \{\mathbf{t}_l\}_{l=k-m}^{k}, \{\mathbf{y}_l\}_{l=k-m}^{k})$

  {linesearch}
  $\{\mathbf{m}_{k+1}, f_{k+1}, \mathbf{g}_{k+1}\} = \mathsf{linesearch}(f_k, \mathbf{g}_k, \delta\mathbf{m}_k)$

  if linesearch successfull **then**
    {update L-BFGS vectors}
    $\mathbf{t}_{k+1} = \mathbf{m}_{k+1} - \mathbf{m}_k$
    $\mathbf{y}_{k+1} = \mathbf{g}_{k+1} - \mathbf{g}_k$

    {draw new sample}
    choose $\mathcal{I}_{k+1} \subseteq \{1, 2, \ldots, M\}$ s.t. $|\mathcal{I}_{k+1}| = b$

    {misfit and gradient for new sample}
    $\{f'_{k+1}, \mathbf{g}'_{k+1}\} = \mathsf{misfit}(\mathbf{m}_{k+1}, \mathcal{I}_{k+1}, \eta)$

    {check average descent}
    if $(f_{k+1} + f'_{k+1}) \geq (f_k + f'_k)$ **then**
      $b = \min(b + \beta, b_{\max})$
    end if
    $f_{k+1} = f'_{k+1}$, $\mathbf{g}_{k+1} = \mathbf{g}'_{k+1}$
    $k = k + 1$
  else
    $\eta = \eta/2$
  end if
end while

To be able to compare the results, we keep track of the residual $||A\mathbf{u}-\mathbf{s}||_2$ (instead of $||(I-Q)\mathbf{u} - R\mathbf{s}||_2$) in CGMN and run GMRES(5) with a right preconditioner. All these experiments are done using the MATLAB native `bicgstab`, `gmres` and `ilu`. For this experiment, we used a naive implementation of CGMN using the explicit expressions for Q and R (cf. (2.4) and (2.5)) to perform the matrix-vector multiplications. As such, we cannot make a fair comparison in terms of CPU time. We focus instead on the number of iterations and ability to converge for higher frequencies.

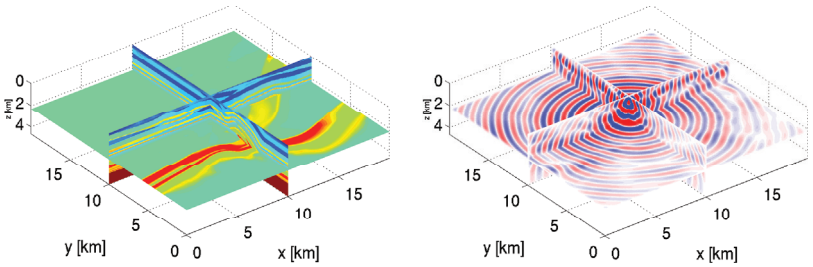


FIG. 1. (a) *Overthrust velocity model,* (b) *wavefield at* 4 *Hz for point source.*

TABLE 1

*Iteration counts for CGMN, BiCGstab, and GMRES for different frequencies (using a constant number of grid points per wavelength). Here, N denotes the total number of unknowns and * indicates that the method did not converge to the desired tolerance of* $\epsilon = 10^{-6}$ *within* 5000 *iterations.*

f [Hz]	N	CGMN	BiCGs	GMRES(5)
0.5	23276.0	308.0	81.0	139.0
1.0	186208.0	564.0	150.0	425.0
2.0	1455808.0	960.0	911.0	1603.0
4.0	11646464.0	2123.0	*	*

The iteration counts for various frequencies (using a constant number of grid points per wavelength) are listed in Table 1. For low frequencies (and hence small systems), BiCGstab converges much faster than both CGMN and GMRES(5). For higher frequencies, however, the difference between the three methods becomes less and for the highest frequency, BiCGstab fails to converge at all. GMRES(5) does well for the low frequencies, but converges very slowly for the highest frequency. This could be countered by increasing the history size, but this would not be feasible in practice due to memory limitations. CGMN, despite being suboptimal for the lower frequencies, does much better than the other two methods for the highest frequency.

The corresponding convergence histories are shown in Figure 2. These plots clearly illustrate the convergence behavior of the different methods; BiCGstab converges very irregularly, and both GMRES and CGMN decrease the residual monotonically.

Block-CG. To test the performance of the block-CG approach, we compute the wavefields for various frequencies for a number of randomly located sources in the $z = 0$ plane using various block sizes for a tolerance of $\epsilon = 10^{-6}$ (in terms of $||(I - Q)\mathbf{u} - R\mathbf{s}||_F$). The results in terms of the number of iterations and CPU times are shown in Table 2.[1] The convergence histories (again in terms of $||(I-Q)\mathbf{u}-R\mathbf{s}||_F$) are shown in Figure 3. We observe that the convergence is not sped up uniformly; for the first 100–200 iterations, all block sizes yield the same result and the largest block sizes show superlinear convergence after this.

The block iterative approach is only beneficial when enough right-hand sides are available, where the optimal number of right-hand sides depends on the size of the system. However, too large block sizes may result in loss of orthogonality of the residuals before the desired tolerance is reached. In these experiments, we restricted

[1]These experiments were done on a dual-core SuperMicro system with 2 Intel Xeon E5-2670@2.6 Ghz and 128 GB RAM.

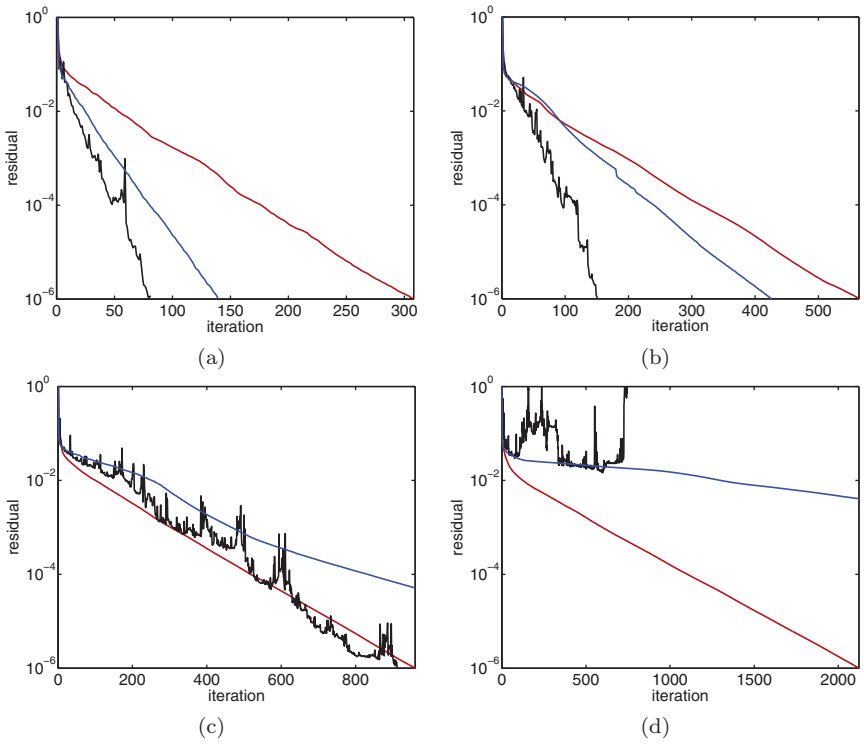


FIG. 2. *Convergence histories for BICGstab (black), GMRES*(5) *(blue), and CGMN (red) for various frequencies:* (a) .5 *Hz,* (b) 1 *Hz,* (c) 2 *Hz, and* (d) 4 *Hz. These plots clearly illustrate the convergence behavior of the different methods; BiCGstab converges very irregulary, and both GMRES and CGMN decrease the residual monotonically.*

TABLE 2

*Iteration counts (averaged over all the runs needed to compute all the wavefields) and CPU times for computing the wavefields for a number of point sources (*10 *for* $f = 0.5$ *Hz,* 50 *for* $f = 1$ *Hz, and* 200 *for* $f = 4$ *Hz) distributed randomly on the surface using different block sizes. Using larger blocks can significantly speed up the convergence, both in terms of the number of iterations and the CPU time.*

f [Hz]	N	block size	iter	time [s]
0.5	23276	1	291	35.9
		2	278	43.3
		5	200	29.7
		10	115	15.2
1.0	186208	1	484	2859.9
		5	477	2419.8
		10	456	2279.7
		50	220	1067.7
2.0	1455808	1	828	125358.2
		10	811	122732.7
		50	716	109424.7
		100	559	82938.2

ourselves to using only modest block sizes thereby circumventing the need for deflation techniques as discussed in section 2.1.

CARP-CG. The domain decomposition is done in the y direction only. Splitting the domain only in the last direction greatly simplifies the implementation, as the

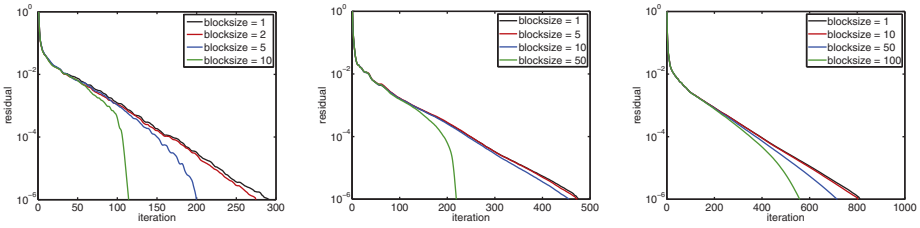


FIG. 3. *Convergence histories for block-CGMN with various block sizes for frequencies* (a) .5 *Hz,* (b) 1 *Hz, and* (c) 2 *Hz. Interestingly, the convergence is not sped up uniformly; for the first* 100–200 *iterations, all block sizes yield the same result. Especially the largest block sizes show superlinear convergence after a certain number of iterations.*

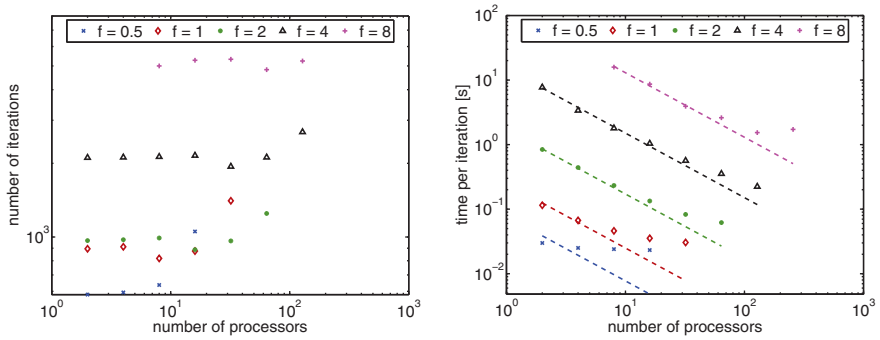


FIG. 4. (a) *Number of iterations as a function of the number of processors for CARP-CG. Ideally, the number of iterations should be independent of the number of domains (processors), but the method becomes slightly less effective when the domains become very small.* (b) *CPU time per iteration as a function of the number of processors for CARP-CG for various frequencies. The dashed line indicates the theoretical CPU time in the case of linear speedup.*

matrix can be split into contiguous parts that have the same band structure as the original matrix. This allows us to use an efficient data structure that only stores the bands of the matrix and the offsets of the bands. As the splitting seems to have little influence on the number of iterations needed to converge, we do not expect that splitting in other or multiple directions will make a big difference.

Figure 4 shows the (a) iteration counts and (b) CPU times[2] for various numbers of processors for a range of frequencies and a tolerance of $\epsilon = 10^{-6}$ (in terms of $||(I - Q)\mathbf{u} - R\mathbf{s}||_2$).

The number of iterations required to converge does not critically increase when using a larger number of subdomains, as can be seen from Figure 4(a) The algorithm achieves nearly linear speedup as can be seen by comparing the actual CPU times with the theoretical times in Figure 4(b).

As the memory imprint of the method is very low, there is no need to scale to thousands of CPUs. Moreover, a higher level of parallelism can be exploited by parallelizing the computation of the misfit and gradient over the sources.

[2]These experiments were done on a cluster with 36 IBM x3550 nodes, each with 2 quad-core 2.6 GHz. Intel CPUs, and 16 GB memory, connected through a Voltaire Infiniband network. Whenever possible we used a maximum of 2 cores per node to avoid cache conflicts. Timings for more than 64 CPUs may therefore be suboptimal.

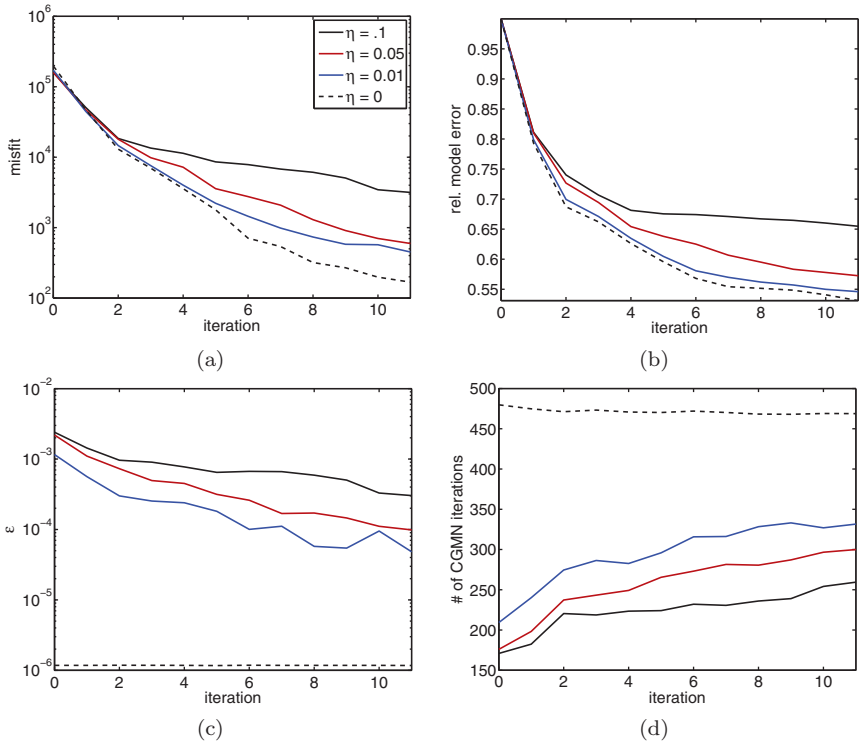


FIG. 5. *Convergence in terms of the* (a) *misfit and* (b) *relative model error for the Edam model. The average (over all sources and frequencies) tolerance used by CGMN for each outer iteration is shown in* (c) *while* (d) *shows the corresponding number of CGMN iterations required. Using more accurate PDE solves (smaller η) yields results closer to the baseline result $\eta = 0$ as can be seen from* (a) *and* (b). *The tolerance used for the PDE solves* (c) *automatically decreases but stays well above the baseline tolerance of 10^{-6}. Consequently, the number of CGMN iterations required is much lower as can be seen in* (d).

4.2. Edam model. We show inversion results on a toy example consisting of a spherical anomaly (resembling an Edam cheese) with a constant velocity of 2500 m/s embedded in a constant background of 2000 m/s. The model is 1 km in each direction and is discretized with 20 m grid spacing and 10 points are added on each side for the PML, leading to a total grid size of $71 \times 71 \times 71$. The "observed" data are generated by solving the Helmholtz equation up to an accuracy of $\epsilon = 10^{-6}$ for 9 sources (located in the $y = 0$-m plane), 2601 receivers (located in the $y = 1000$-m plane) and 3 frequencies of 5, 10, and 15 Hz.

For the inversion we use all sources and frequencies simultaneously and conduct three experiments using $\eta = \{0.1, 0.05, 0.01\}$ and compare these to a baseline result obtained using very accurate PDE solves ($\eta = 0$). The data misfit at each iteration is shown in Figure 5(a), the relative model error is shown in Figure 5(b), the tolerance used for the PDE solves is shown in Figure 5(c), and the number of CGMN iterations required to reach that tolerance is shown in Figure 5(d). The tolerance gradually decreases (automatically), confirming that very accurate PDE solves are not needed in the early stages of the inversion. Moreover, the results for $\eta = 0.01$ are nearly identical to those for $\eta = 0$, while the former was roughly twice as cheap, as can be seen by computing the total number of CGMN iterations needed from Figure 5(d). The true and reconstructed models are shown in Figure 6. All the reconstructions

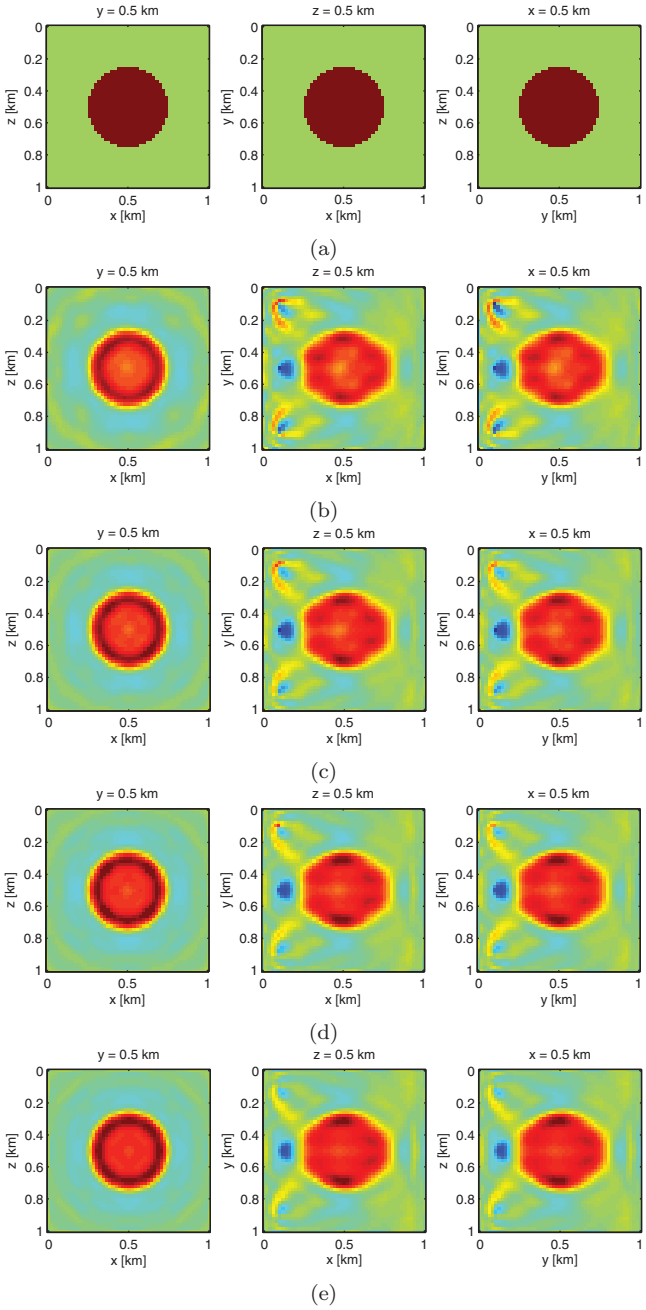


FIG. 6. *The true velocity model used for the Edam experiment is shown in* (a). *The reconstructed models using* (b) $\eta = 0.1$, (c) $\eta = 0.05$, (d) $\eta = 0.01$, *and* (e) $\eta = 0$ *are also shown on the same color scale. All the reconstructions are very reasonable when compared to the baseline result* (e). *Using more accurate solves* $\eta = 0.01$ (d) *yields fewer artifacts and is almost identical to the baseline result, however, the computational cost of the baseline was roughly twice as high.*

are very reasonable when compared to the baseline result (Figure 6(e)). Using more accurate solves $\eta = 0.01$ (Figure 6(d)) yields fewer artifacts and is almost identical to the baseline result, however, the computational cost of the baseline result was roughly twice as high.

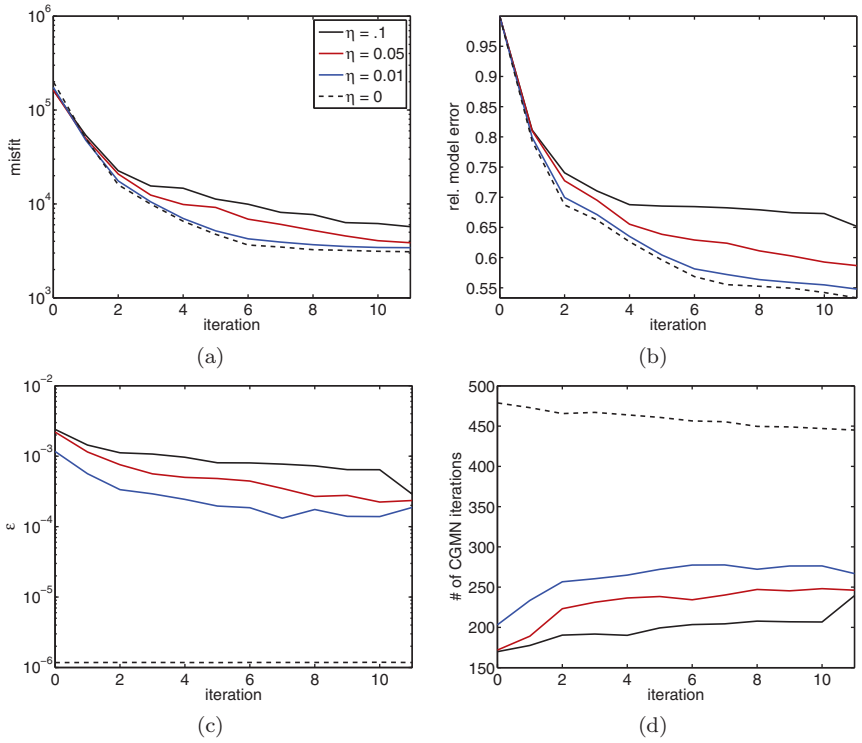


FIG. 7. *Convergence in terms of the* (a) *misfit and* (b) *relative model error for the Edam model using noisy data (5% noise). The average (over all sources and frequencies) tolerance used by CGMN for each outer iteration is shown in* (c) *while* (d) *shows the corresponding number of CGMN iterations required. Using more accurate PDE solves (smaller η) yields results closer to the baseline result $\eta = 0$ as can be seen from* (a) *and* (b). *The tolerance used for the PDE solves* (c) *automatically decreases but stays well above the baseline tolerance of 10^{-6}. Consequently, the number of CGMN iterations required is much lower as can be seen in* (d).

We repeat the above described experiment using data with 5% and 10% additive Gaussian noise. The convergence plots are shown in Figures 7 and 8, respectively. Comparing these figures to the noiseless case (Figure 5) we see that the method is not affected by this moderate amount of noise.

4.3. Overthrust model. We generate data using iWAVE, an open-source time-domain finite-difference code [45], for a 5 km x 5 km central part of a well-known benchmark model for seismic inversion (the Overthrust model) gridded with 50 m spacing. The true and initial models are shown in Figure 9. A total of 121 sources and 2601 receivers (both regularly spaced) cover the top of the model.

While significantly bigger than the previous problem, this is still a postage stamp model compared to industry-scale problems, which easily consist of 10^9 grid points and thousands of sources.

Experiment 1. To show the influence of the sample size on the optimization, we invert single-frequency data at 4 Hz using a grid spacing of 100 m and five points for the PML, leading to a gridsize of $36 \times 61 \times 61$. We use various constant sample sizes and compare this to using a growing sample size. The convergence histories in terms of both the data misfit and model error as a function of the effective number

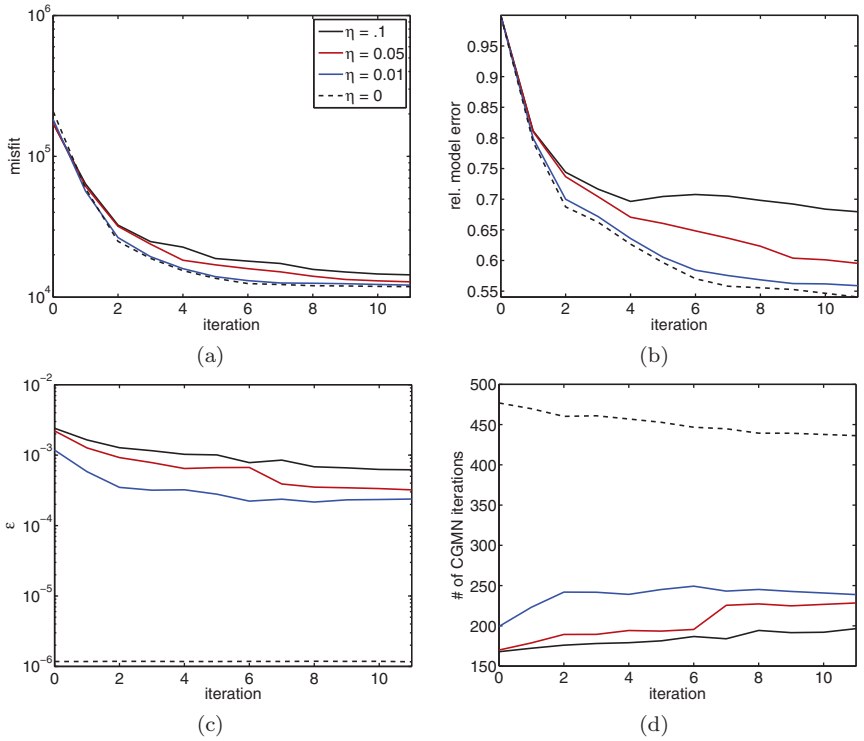


FIG. 8. *Convergence in terms of the* (a) *misfit and* (b) *relative model error for the Edam model using noisy data (*10% *noise). The average (over all sources and frequencies) tolerance used by CGMN for each outer iteration is shown in* (c) *while* (d) *shows the corresponding number of CGMN iterations required. Using more accurate PDE solves (smaller* η*) yields results closer to the baseline result* $\eta = 0$ *as can be seen from* (a) *and* (b). *The tolerance used for the PDE solves* (c) *automatically decreases but stays well above the baseline tolerance of* 10^{-6}. *Consequently, the number of CGMN iterations required is much lower as can be seen in* (d).

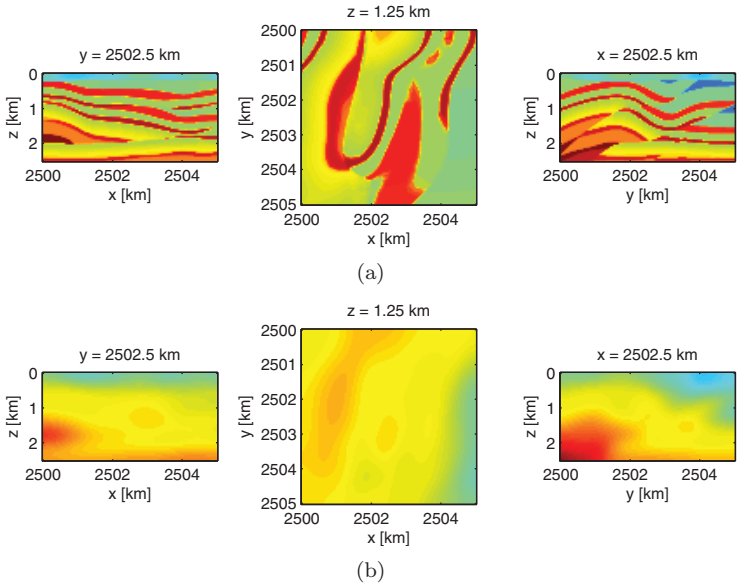


FIG. 9. *A central part of the Overthrust model is shown in* (a). *The initial model, which is a smoothed version of the true model, used for the inversion is shown in* (b) *on the same color scale.*

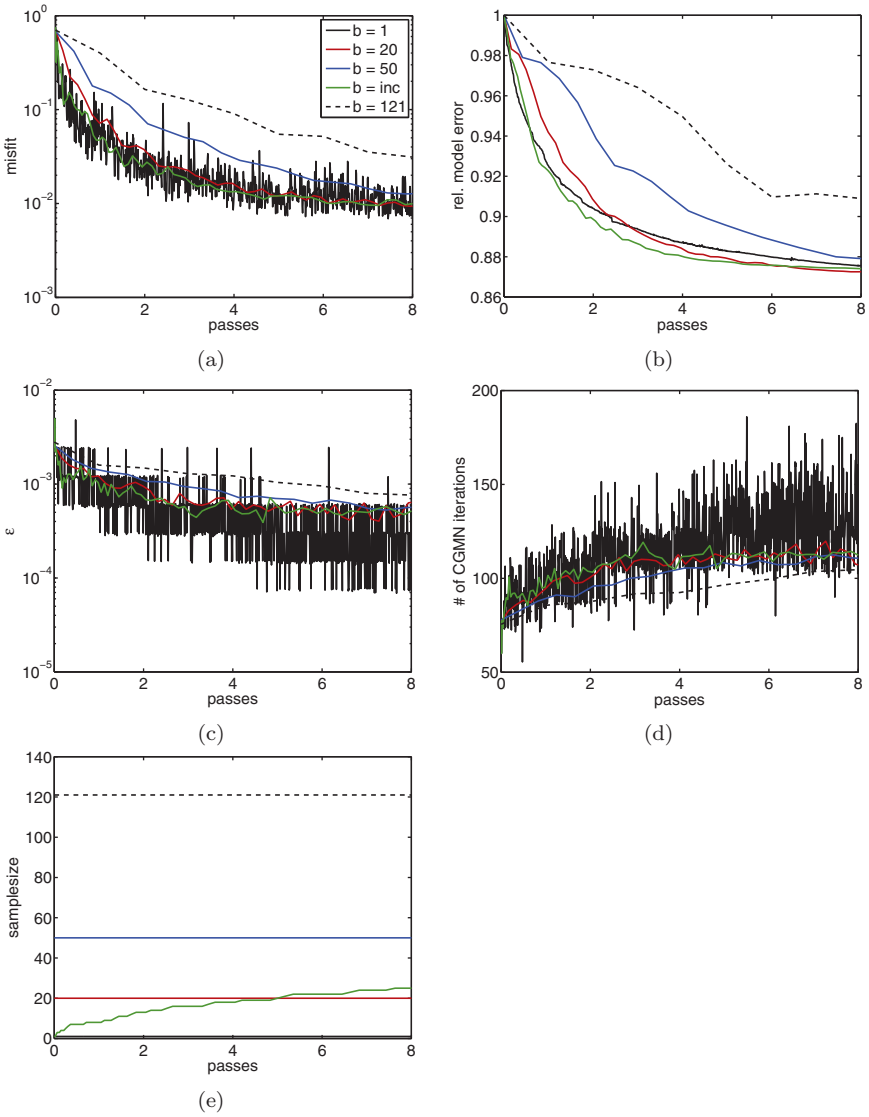
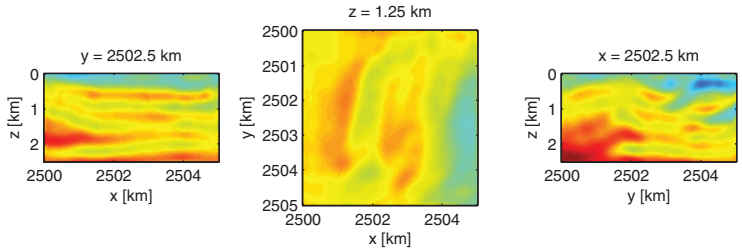
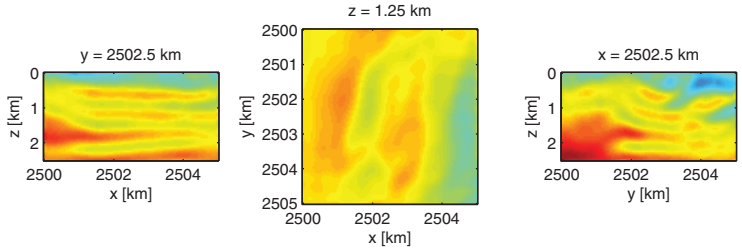


FIG. 10. *Convergence in terms of the* (a) *misfit and* (b) *relative model error for the first Overthrust experiment. The average (over all sources) tolerance used by CGMN for each outer iteration is shown in* (c) *while* (d) *shows the corresponding number of CGMN iterations required. All results are shown as a function of the effective number of complete passes through the data, and hence have the same computational cost. The sample size is shown in* (e).

of passes through the data, are shown in Figures 10(a) and (b). All the results have the same computational cost, however, we can perform more outer iterations when using smaller sample sizes. All subsampled cases do much better than the case where we use all the data. Remarkably, even using just a single source at each iteration does very well. Using $b = 20$ sources does less well in the beginning but eventually catches up to the $b = 1$ case. Increasing the sample size does slightly better than both $b = 1$ and $b = 20$, at least up to 6 passes through the data when all three yield the same model error. Looking at the used tolerance at each iteration in Figure 10(c), we see that there is trade-off; using a bigger sample size results in a higher tolerance for the PDE solves. However, the effect is fairly small and looking at the required



(a)



(b)

FIG. 11. *Reconstructed models using* (a) $b_0 = 1$ (a) *and* (b) $b_0 = 121$. *Although the reconstructions look similar at a first glance, closer inspection reveals that the results for* $b = 1$ *have slightly more detail.*

number of CGMN iterations in Figure 10(d), we see that they all require roughly the same number of iterations. Finally, the used sample size is depicted in Figure 10(e), showing that even when increasing the sample size, we never use all the sources in a given iteration. The resulting reconstructions using $b = 1$ and $b = 121$ are shown in Figure 11. Although the reconstructions look similar at a first glance, closer inspection reveals that the result for $b = 1$ has slightly more detail. For this experiment, we do not expect to see large differences between the different approaches because we run a large number of outer iterations, even using all the sources ($b = 121$). We would expect more dramatic differences when doing only a few passes though the data, which is what we do in the next experiment.

Experiment 2. To illustrate the multiscale inversion approach, we invert frequencies $f = 4, 6, 8$ Hz consecutively using a grid spacing of 100 m, 66.67 m, and 50 m, respectively, all with five points for the PML. This leads to total grid sizes of $36 \times 61 \times 61$, $48 \times 86 \times 86$, and $61 \times 111 \times 111$. We use either a fixed number of $b = 1$ and $b = 121$ sources or an increasing number of sources, starting from $b = 1$. For all cases we perform two passes through the data for each frequency. The convergence histories in terms of both the misfit and the model error are shown in Figures 12(a) and (b). The sudden jumps in the misfit are caused by switching from one frequency to the next. The stochastic method outperforms the regular approach that uses all the sources. Moreover, gradually increasing the sources yields a better result than using only one source at each iteration. The tolerances and corresponding number of CGMN iterations are shown in Figures 12(c) and (d). Again, the tolerance decreases gradually as the iterations proceed. The corresponding number of CGMN iterations increases mildly, partly because the tolerance decreases but also because the grid is refined when going to a higher frequency. Figure 12(d) shows the sample size. The increasing strategy never uses more than a small fraction of the total number of sources. Finally, the reconstructed models are shown in Figure 13. The difference in recon-

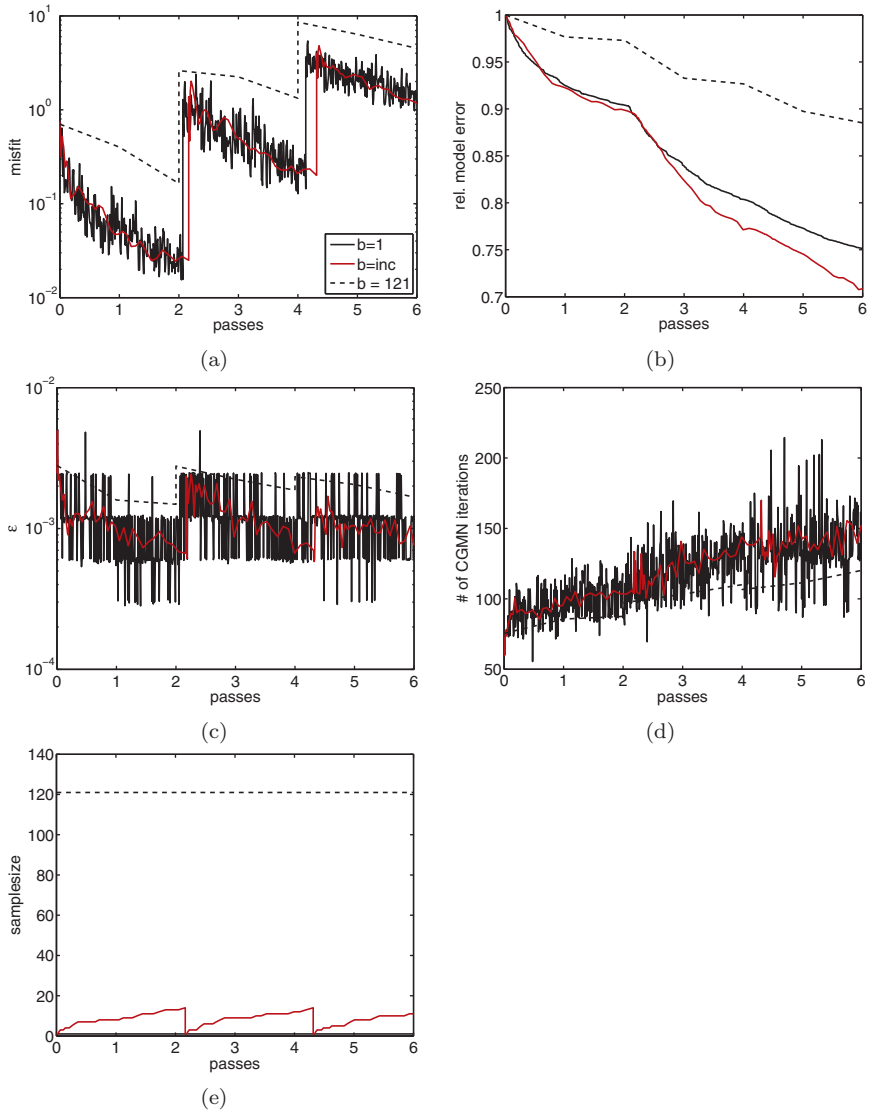


FIG. 12. *Convergence in terms of the* (a) *misfit and* (b) *relative model error for the second Overthrust experiment. The jumps in the misfit are caused by switching to a higher frequency. The average (over all sources) tolerance used by CGMN for each outer iteration is shown in* (c) *while* (d) *shows the corresponding number of CGMN iterations required. All results are shown as a function of the effective number of complete passes through the data, and hence have the same computational cost. The sample size is shown in* (e).

struction between using all the sources and only a few is very clear; both Figures 13(a) and (c) show much more detail than Figure 13(b).

To obtain a comparable result using all the sources (i.e., $b = 121$) we needed 10 passes through the data for each frequency band. In this case, the stochastic method is five times faster than a conventional approach.

5. Conclusions and discussion. In this paper, we outlined the challenges faced by industry to come up with a versatile and practical 3D inversion scheme that scales to large problem sizes with many right-hand sides. We made it clear that coming

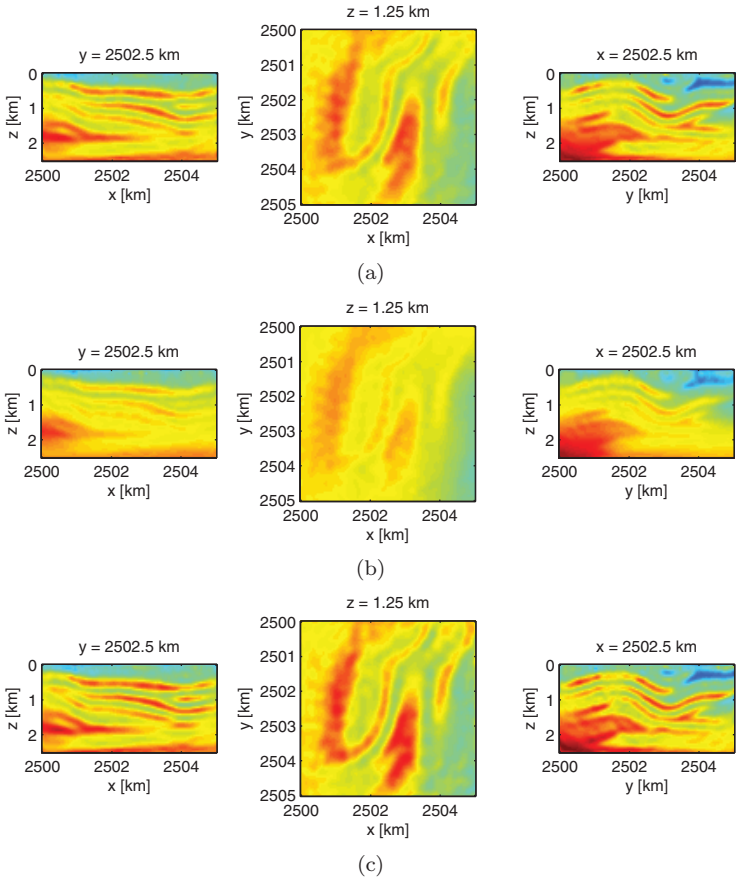


FIG. 13. *Reconstructed models using a fixed sample size of* (a) $b_0 = 1$ *and* (b) $b_0 = 121$, *and* (c) *an increasing sample size, starting from* $b_0 = 1$. *The difference in reconstruction between using all the sources and only a few is very clear. Both* (a) *and* (c) *show much more detail than* (b).

up with scalable inversion schemes is extremely challenging because of (i) the need to propagate tens to hundreds of wavelenghts, leading to large computational grids ($\sim 10^9$ grid points), and (ii) the large amount of source experiments (typically thousands), requiring thousands of PDE solves at each iteration of a conventional nonlinear data-fitting algorithm. While industry is slowly incorporating FWI based on the scalar wave equation into their workflows, major challenges remain given the recent push towards multiparameter inversions with vector Helmholtz equations.

We address a number of these challenges by developing a new stochastic inversion algorithm designed to work with small subsets of right-hand sides and inexact PDE solves. Our method derives from the observation that we can allow for large errors in the gradient calculations (as compared to the full gradient computed with exact PDE solves and all the right-hand sides) in the course of a gradient-based optimization algorithm, as long as these errors are gradually decreased as the algorithm converges. The errors are controlled via heuristics that automatically adapt the sample size and accuracy of the PDE solves.

To solve the Helmholtz equation we used the CGMN method and its parallel extension, CARP-CG. While this preconditioner may not be optimal in the sense of

requiring only a few iterations or an iteration count that is independent of the size of the system, our approach has a few notable advantages that make it a very suitable candidate for use in a scalable inversion framework, namely, (i) it is guaranteed to converge and does not rely on model-specific tuning parameters that may change as the inversion progresses, (ii) the iterations are very cheap, are easily parallelized, and allow for straightforward matrix-free implementation, (iii) the preconditioner is generic and hence the inversion algorithm can be easily extended to multiparameter elastic inversion using a vector Helmholtz equation. Aside from these three key advantages, the CARP-CG-based preconditioner is well suited for a control of the accuracy during the early stages of the inversion. For instance, in our examples we never needed very accurate solves limiting the Helmholtz solves to no more than a few hundred CGMN iterations to converge to the desired accuracy. We also found in our numerical examples that CGMN outperforms BiCGstab and GMRES (both with an ILU(0) preconditioner) for high frequencies.

To further speed up the inversion of multisource seismic data, we extended CGMN to a block iterative method that handles multiple right-hand sides at the same time. Numerical experiments showed that this block iterative Helmholtz solver can significantly speed up the convergence as long as we avoid a breakdown due to loss of orthogonality of the residuals. In our application the right-hand sides represent individual source experiments, which are naturally independent when their physical locations are far apart. Hence, the initial residuals are naturally orthogonal and in our numerical experiments we typically reach the desired tolerance before the residuals become (numerically) linearly dependent. For a more robust implementation, variable-block or deflation techniques can be used but these are beyond the scope of this paper. Finally, the parallel extension of the Kaczmarz-based preconditioner we used resembles an additive Schwarz approach but with the difference that convergence is still guaranteed. Moreover, our numerical examples show that the method scales well and that using more domains does not critically slow down the convergence.

Aside from heuristics to adapt sample size and accuracy as needed by the inversion, the true contribution of this paper lies in striking a balance between speeding up the Helmholtz solves, via model-space parallelism in the form of domain decomposition, and data-space parallelism by looping over different frequencies and source subsets. Striking this balance is critical because massively parallel Helmholtz solves, with their memory overhead and setup costs, would immediately saturate the largest compute clusters for realistic FWI problems that involve many source experiments. Without the high level of parallelism via domain decompositions, blocks of multiple right-hand sides, and parallelism over randomized source subsets and frequencies, we would not be able to attain the level of parallelism required by FWI in an industrial setting.

We demonstrate the efficacy of our inversion framework on two relatively small benchmark models (132651 unknowns, 9 sources and 520251 unknowns, 121 sources) and the results are very encouraging and show the practical validity of the proposed heuristics of adaptively increasing the accuracy of the PDE solves and the sample size. With our adaptive approach, we get competitive inversion results at a cost that is equivalent to performing only a few iterations with *all* the data at full accuracy.

In summary, we developed an adaptive stochastic algorithm for seismic waveform inversion geared towards application to large-scale seismic datasets. We attain a high level of parallelism by iteratively inverting domain-decomposed Helmholtz systems for manageable numbers of blocked right-hand sides made of randomized subsets of source experiments. Our approach leads to drastic accelerations as it only increases

the sample size and accuracy of the Helmholtz solves as needed by the optimization to converge. Numerical results on small benchmark models are very promising. While scalability to industrial data sets has not been shown, we see no fundamental difficulties in doing so.

Acknowledgments. The authors wish to thank Dan and Rachel Gordon and Michael Friedlander for valuable discussions, and Uri Ascher and Eldad Haber for valuable feedback on the first draft of this manuscript. Zhi Long Fang is gratefully acknowledged for his contribution to the parallel MATLAB code.

REFERENCES

[1] A.Y. Aravkin, M.P. Friedlander, F.J. Herrmann, and T. van Leeuwen, *Robust inversion, dimensionality reduction, and randomized sampling*, Math. Program., 134 (2012), pp. 101–125.

[2] A.Y. Aravkin and T. van Leeuwen, *Estimating nuisance parameters in inverse problems*, Inverse Problems, 28 (2012), 115016.

[3] M. Arioli, I. S. Duff, D. Ruiz, and M. Sadkane, *Block Lanczos techniques for accelerating the block Cimmino method*, SIAM J. Sci. Comput., 16 (1995), pp. 1478–1511.

[4] G. Biros and O. Ghattas, *Inexactness issues in the Lagrange-Newton-Krylov-Schur method for PDE-constrained optimization*, in Large-Scale PDE-Constrained Optimization, Springer, Berlin, 2003, pp. 93–114.

[5] Å. Björck and T. Elfving, *Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations*, BIT, 19 (1979), pp. 145–163.

[6] C. Bunks, *Multiscale seismic waveform inversion*, Geophysics, 60 (1995), pp. 1457–1473.

[7] R.H. Byrd, G.M. Chin, W. Neveitt, and J. Nocedal, *On the use of stochastic Hessian information in optimization methods for machine learning*, SIAM J. Optim., 21 (2011), pp. 977–995.

[8] H. Calandra, S. Gratton, J. Langou, X. Pinel, and X. Vasseur, *Flexible variants of block restarted GMRES methods with application to geophysics*, SIAM J. Sci. Comput., 34 (2012), pp. A714–A736.

[9] B. Engquist, *Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation*, Comm. Pure Appl. Math., 64 (2011), pp. 697–735.

[10] I. Epanomeritakis, V. Akçelik, O. Ghattas, and J. Bielak, *A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion*, Inverse Problems, 24 (2008), 034015.

[11] Y.A. Erlangga, C. Vuik, and C.W. Oosterlee, *On a robust iterative method for heterogeneous Helmholtz problems for geophysics applications*, Int. J. Numer. Anal. Model., 2 (2005), pp. 197–208.

[12] O.G. Ernst and M.J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, in Numerical Analysis of Multiscale Problems, I.G. Graham, T.Y. Hou, O. Lakkis, and R. Scheichl, eds., Lect. Notes Comput. Sci. and Eng. 83, Springer, Berlin, 2012, pp. 325–363.

[13] M.P. Friedlander and M. Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM J. Sci. Comput., 34 (2012), pp. A1380–A1405.

[14] D. Gordon and R. Gordon, *Component-averaged row projections: A robust, block-parallel scheme for sparse linear systems*, SIAM J. Sci. Comput., 27 (2005), pp. 1092–1117.

[15] D. Gordon and R. Gordon, *CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs*, Parallel Comput., 36 (2010), pp. 495–515.

[16] D. Gordon and R. Gordon, *Parallel solution of high-frequency Helmholtz equations using high-order finite difference schemes*, Appl. Math. Comput., 218 (2012), pp. 10737–10754.

[17] D. Gordon and R. Gordon, *Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers*, J. Comput. Appl. Math., 232 (2012), pp. 182–196.

[18] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Front. Appl. Math. 17, SIAM, Philadelphia, 1997.

[19] E. Haber, U.M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.

[20] E. Haber, M. Chung, and F. Herrmann, *An effective method for parameter estimation with PDE constraints with multiple right-hand sides*, SIAM J. Optim., 22 (2012), pp. 739–757.

[21] E. Haber and S. MacLachlan, *A fast method for the solution of the Helmholtz equation*, J. Comput. Phys., 230 (2011), pp. 4403–4418.
[22] M. Heinkenschloss and D. Ridzal, *An inexact trust-region SQP method with applications to PDE-constrained optimization*, in Numerical Mathematics and Advanced Applications, Springer, Berlin, 2008, pp. 613–620.
[23] M. Heinkenschloss and L.N. Vicente, *Analysis of inexact trust-region SQP algorithms*, SIAM J. Optim., 12 (2002), pp. 283–302.
[24] S. Kaczmarz, *Angenäherte auflösung von systemen linearer gleichungen*, Bull. Int. Acad. Polonaise Sci. Lett., 35 (1937), pp. 355–357.
[25] H. Knibbe, W.A. Mulder, C.W. Oosterlee, and C. Vuik, *Closing the performance gap between an iterative frequency-domain solver and an explicit time-domain scheme for 3D migration on parallel architectures*, Geophysics, 79 (2014), pp. 547–561.
[26] X. Li, A.Y. Aravkin, T. van Leeuwen, and F.J. Herrmann, *Fast randomized full-waveform inversion with compressive sensing*, Geophysics, 77 (2012), pp. A13–A17.
[27] Y. Luo and G. Schuster, *Wave-equation traveltime inversion*, Geophysics, 56 (1991), pp. 645–653.
[28] P.P. Moghaddam, H. Keers, F.J. Herrmann, and W.A. Mulder, *A new optimization approach for source-encoding full-waveform inversion*, Geophysics, 78 (2013), pp. R125–R132.
[29] A.A. Nikishin and A.Yu. Yeremin, *Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers,* I: *General iterative scheme*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1135–1153.
[30] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, 2000.
[31] S. Operto, J. Virieux, P. Amestoy, J.Y. L'Excellent, L. Giraud, and H.B.H. Ali, *3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study*, Geophysics, 72 (2007), pp. SM195–SM211.
[32] D. Osei-Kuffuor and Y. Saad, *Preconditioning Helmholtz linear systems*, Appl. Numer. Math., 60 (2010), pp. 420–431.
[33] R.-E. Plessix, *A review of the adjoint-state method for computing the gradient of a functional with geophysical applications*, Geophys. J. Int., 167 (2006), pp. 495–503.
[34] R.-E. Plessix, Y.-H. De Roeck, and G. Chavent, *Waveform inversion of reflection seismic data for kinematic parameters by local optimization*, SIAM J. Sci. Comput., 20 (1999), pp. 1033–1052.
[35] J. Poulson, B. Engquist, S. Fomel, S. Li, and L. Ying, *A Parallel Sweeping Preconditioner for High Frequency Heterogeneous 3D Helmholtz Equations*, preprint, arXiv:1204.0111, 2012.
[36] G.R. Pratt, C. Shin, and G.J. Hicks, *Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion*, Geophys. J. Int., 133 (1998), pp. 341–362.
[37] R.G. Pratt, Z.M. Song, P. Williamson, and M. Warner, *Two-dimensional velocity models from wide-angle seismic data by wavefield inversion*, Geophys. J. Int., 124 (1996), pp. 232–340.
[38] C. Riyanti, A. Kononov, Y.A. Erlangga, C. Vuik, C. Oosterlee, R.-E. Plessix, and W.A. Mulder, *A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation*, J. Comput. Phys., 224 (2007), pp. 431–448.
[39] N.N. Schraudolph, J. Yu, and S Günter, *A stochastic Quasi-Newton method for online convex optimization*, in Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico, 2007, pp. 436–443.
[40] F. Sourbier, A. Haidar, L. Giraud, H. Ben-Hadj-Ali, J. Virieux, and S. Operto, *Three-dimensional parallel frequency-domain visco-acoustic wave modelling based on a hybrid direct/iterative solver*, Geophys. Prospect., 59 (2011), pp. 834–856.
[41] W.W. Symes, *Layered velocity inversion: A model problem from reflection seismology*, SIAM J. Math. Anal., 22 (1991), pp. 680–716.
[42] W.W. Symes, *Reverse time migration with optimal checkpointing*, Geophysics, 72 (2007), pp. SM213–SM221.
[43] W.W. Symes, *The seismic reflection inverse problem*, Inverse Problems, 25 (2009), 123008.
[44] A. Tarantola and A. Valette, *Generalized nonlinear inverse problems solved using the least squares criterion*, Rev. Geophys. Space Phys., 20 (1982), pp. 129–232.
[45] I.S. Terentyev, T Vdovina, W.W. Symes, X. Wang, and D. Sun, *IWAVE: A Framework for Wave Simulation*, http://www.trip.caam.rice.edu/software/iwave/doc/html/ (2010).
[46] K. van den Doel and U.M. Ascher, *Adaptive and stochastic algorithms for electrical impedance tomography and DC resistivity problems with piecewise constant solutions and many measurements*, SIAM J. Sci. Comput., 34 (2012), pp. A185–A205.

[47] T. van Leeuwen, *Fourier Analysis of the CGMN Method for Solving the Helmholtz Equation*, preprint, arXiv:1210.2644, 2012.
[48] T. van Leeuwen, A.Y. Aravkin, and F.J. Herrmann, *Seismic waveform inversion by stochastic optimization*, Int. J. Geophys., 2011 (2011), 689041.
[49] T. van Leeuwen and F.J. Herrmann, *Fast waveform inversion without source-encoding*, Geophys. Prospect., 61, Suppl. (2012), pp. 10–19.
[50] T. van Leeuwen and F.J. Herrmann, *Mitigating local minima in full-waveform inversion by expanding the search space*, Geophys. J. Int., 195 (2013), pp. 661–667.
[51] T. van Leeuwen and W.A. Mulder, *A comparison of seismic velocity inversion methods for layered acoustics*, Inverse Problems, 26 (2010), 015008.
[52] T. van Leeuwen and W.A. Mulder, *A correlation-based misfit criterion for wave-equation traveltime tomography*, Geophys. J. Int., 182 (2010), pp. 1383–1394.
[53] J. Virieux and S. Operto, *An overview of full-waveform inversion in exploration geophysics*, Geophysics, 74 (2009), pp. WCC1–WCC26.
[54] S. Wang, M.V. de Hoop, and J. Xia, *On 3D modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver*, Geophys. Prospect., 59 (2011), pp. 857–873.
[55] J. Young and D. Ridzal, *An application of random projection to parameter estimation in partial differential equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2344–A2365.