Parallel reformulation of the sequential adjoint-state method

*Bas Peters[*1], Tristan van Leeuwen[2], Felix J. Herrmann[1]*
[1]*Seismic Laboratory for Imaging and Modeling (SLIM), University of British Columbia.*
[2]*Mathematical Institute, Utrecht University.*

SUMMARY

This abstract is about reducing the total computation time to solve full-waveform inversion problems. A common problem formulation is a nonlinear least-squares problem where the gradient is computed using the adjoint-state algorithm. This formulation offers parallelism for the computation of gradients for different sources and frequencies. The adjoint-state algorithm itself is sequential however and this is a limiting factor when a lot of compute nodes are available and only a few wavefields need to be computed. This situation occurs when stochastic optimization strategies are used to minimize the objective function. We present a parallel reformulation of the sequential adjoint-state algorithm, which allows the forward- and adjoint wavefields to be computed in parallel. Both algorithms are mathematically equivalent but the parallel version is twice as fast in run time. An important characteristic of the proposed algorithm is that one wavefield needs to be computed per source and one per receiver. These fields can be used to apply the (inverse) Gauss-Newton Hessian to a vector without recomputing wavefields. A 2D example shows that good full-waveform inversion results are obtained, even when a small number of sources and receivers is used.

INTRODUCTION

This work is about the computational aspects of frequency-domain full-waveform inversion (FWI). The most costly computational component of full-waveform inversion is the solution of partial differential equations (PDEs), in acoustic inversions this is the scalar Helmholtz equation. Simple gradient descent and quasi-Newton algorithms to minimize a nonlinear data-misfit function require gradients and function values, which in turn require PDE solves. In this work, it is assumed that the PDEs are solved using iterative methods.

There are several ways to reduce the computational cost. Faster PDE solvers or optimization algorithms with a better rate of convergence is one option. Other work focusses on using a smaller number of (simultaneous) sources per FWI iteration, utilizing randomization and subsampling techniques (Krebs et al., 2009,Symes (2010), Habashy et al. (2011), Haber et al. (2012), van Leeuwen and Herrmann (2013), van Leeuwen et al. (2011)).

In this work, we focus on the problem formulation itself to achieve a speedup in time, given fixed computational resources. We also show that the proposed reformulation is very well suited to work with source and receiver randomization and subsampling.

We will follow the common formulation of full-waveform inversion as the solution of a nonlinear least-squares problem. This formulation has been used by many authors over the past few decades, together with the adjoint-state technique to compute the gradient of of the nonlinear least-squares objective function. Although wavefields can be computed in parallel for different sources and different frequencies, the adjoint-state algorithm itself is sequential. A forward wavefield is solved first, the resulting data-residual is then back-propagated using another PDE solve. However, the adjoint-state sequence of operations is just one way to compute a gradient for the full-waveform inversion problem. In this work, we show that a parallel (as opposed to sequential) reformulation is also possible, yielding a $2\times$ speedup in run time.

A numerical example illustrates the feasibility of this method.

PARALLELIZING FORWARD AND ADJOINT SOLVES

Consider the following widely used objective function for frequency-domain full-waveform inversion (FWI):

$$f(\mathbf{m}) = \frac{1}{2}\sum_{i=1}^{n_\mathrm{s}} \|PA(\mathbf{m})^{-1}\mathbf{q}_i - \mathbf{d}_i\|_2^2, \tag{1}$$

where the goal is to find a local minimizer. The matrix $A(\mathbf{m}) \in \mathbb{C}^{N\times N}$ is the Helmholtz operator for a single frequency, discretized on a $n_1 \times n_2 \times n_3 = N$ grid depending on the medium parameters $\mathbf{m} \in \mathbb{R}^N$. The operator $P \in \mathbb{R}^{n_\mathrm{r}\times N}$ restricts the wavefield to the receiver locations ('observation' matrix), $\mathbf{q}_i \in \mathbb{C}^N$ is the vector containing the source function and $\mathbf{d}_i \in \mathbb{C}^{n_\mathrm{r}}$ is the observed data. A well-known technique to compute the gradient $\mathbf{g} = \nabla_\mathbf{m} f(\mathbf{m})$ is shown in Algorithm 1 below. The complex-conjugate transpose in this pseudo code is indicated by a star *. The partial derivative $\partial_\mathbf{m}(A(\mathbf{m})\mathbf{u})$ is given by $\omega^2\mathrm{diag}(\mathbf{u})$ in case the Helmholtz equation is discretized in the form $A(\mathbf{m}) = L + \omega^2\mathrm{diag}(\mathbf{m})$ with the discrete Laplacian L. The term $\mathrm{diag}(\mathbf{m})$ is interpreted as a diagonal matrix with $\mathbf{m}$ on the diagonal. n_s indicates the number of sources and n_r the number of receivers.

Algorithm 1 The conventional sequential adjoint-state algorithm to compute $\mathbf{g}$.

1. $\mathbf{u}_i = A(\mathbf{m})^{-1}\mathbf{q}_i$ `//forward solve`
2. $\mathbf{v}_i = A(\mathbf{m})^{-*}(P^*(P\mathbf{u}_i - \mathbf{d}_i))$ `//adjoint solve`
3. $\mathbf{g}_i = \left(\frac{\partial A(\mathbf{m})\mathbf{u}_i}{\partial_\mathbf{m}}\right)^* \mathbf{v}_i$ `//evaluate gradient`
4. $\mathbf{g} = \sum_{i=1}^{n_\mathrm{s}} \mathbf{g}_i$ `//sum gradient components`

Although the above gradient can be computed for every source and frequency in parallel, the adjoint-state algorithm itself is essentially sequential. The forward problem needs to be solved first before the adjoint-computation can be started. If the parallel computational resources are such that more Helmholtz

problems can be solved in parallel than there are forward or adjoint problems, the adjoint-state algorithm cannot take advantage of these available resources.

As we mentioned earlier, the adjoint-state algorithm (Algorithm 1) is just one instance of the several options to compute the gradient $\nabla_{\mathbf{m}} f(\mathbf{m})$. For instance, a different algorithm can be obtained by first substituting the forward equation into the adjoint wave equation, i.e., $\mathbf{v}_i = A(\mathbf{m})^{-*}(P^*(PA(\mathbf{m})^{-1}\mathbf{q}_i - \mathbf{d}_i))$ and then factoring out the term $W = A(\mathbf{m})^{-*}P^*$. The adjoint wavefield can now be written as $\mathbf{v}_i = W(PA(\mathbf{m})^{-1}\mathbf{q}_i - \mathbf{d}_i))$. If the dense matrix $W \in \mathbb{C}^{N \times n_r}$ is computed explicitly, the computations of $\mathbf{u}$ and W are independent and can be done in parallel. Moreover, there are no Helmholtz solves required to obtain the adjoint wavefield in this case, just matrix-vector products with the dense matrix W and vector additions are required. The computation of W requires one adjoint Helmholtz solve per receiver. Because we solve 1 or a few Helmholtz problems on each compute node, W is a column distributed matrix.

This alternative algorithm to compute the gradient of (1) is summarized in Algorithm 2 and some important properties are listed in Table 1 as well.

Algorithm 2 The parallel reformulation to compute $\mathbf{g}$.

1. $\mathbf{u}_i = A(\mathbf{m})^{-1}\mathbf{q}$ & $W = A(\mathbf{m})^{-*}P^*$ `//solve in parallel`
2. $\mathbf{v}_i = -W(P\mathbf{u}_i - \mathbf{d}_i))$ `//evaluate adjoint`
3. $\mathbf{g}_i = \left(\frac{\partial A(\mathbf{m})\mathbf{u}_i}{\partial_{\mathbf{m}}}\right)^* \mathbf{v}_i$ `//evaluate gradient`
4. $\mathbf{g} = \sum_{i=1}^{n_s} \mathbf{g}_i$ `//sum gradient components`

The parallel reformulation of Algorithm 2 can take advantage in cases where many compute nodes are available and where we work with relatively few sources and receivers. In that case, our parallelized gradient algorithm runs at twice the speed in terms of total time compared to the conventional adjoint-state method (Algorithm. 1) if $n_s + n_r \leq$ (number of Helmholtz problems that can be solved in parallel). Because most seismic surveys involve hundreds, if not thousands or more sources and receivers, we can only realize this factor 2 speedup in a stochastic optimization framework where only a few sources and receivers are used at every iteration of FWI. While the conventional adjoint-state method has the advantage that receivers come for 'free', not all receivers are required at each iteration when a stochastic optimization framework is used. This is illustrated with an example in a later section.

STOCHASTIC OPTIMIZATION VIA SOURCE/RECEIVER RANDOMIZED SUBSAMPLING

The main idea is to minimize the separable objective $f(\mathbf{m})$ by working with only a few of it's components, i.e. (composite) sources and/or (composite) receivers, per iteration. These components can be changed every iteration or after a few iterations. In this way, all sources and receivers can be touched as the iterations progress. (Haber et al., 2012,van Leeuwen and Herrmann (2013), van Leeuwen et al. (2011))

In this application of stochastic optimization ideas, we redraw at every iteration a new set of randomly selected sources and receivers. Algorithm 2 also works in conjunction with other randomization and subsampling techniques such as simultaneous sources and methods based on singular value decomposition of data or data-misfit matrices (Symes, 2010,Habashy et al. (2011)). van Leeuwen and Herrmann (2013) and van Leeuwen et al. (2011) compare different subsampling strategies.

For our purposes, using a smaller number of sources for each iteration is not enough, because Algorithm 2 requires one PDE solve per receiver. Therefore some kind of receiver subsampling is also required. Receiver compression was used before by Habashy et al. (2011) in the context of simplifying the computations required to apply the Gauss-Newton Hessian corresponding to the objective (1) in a serial computing setting. We randomly draw subsets of receivers.

Working with subsets of sources and receivers results in approximations to the objectives $\tilde{f}(\mathbf{m})$ and gradients $\tilde{\mathbf{g}}$, compared to the full objective based on all sources and receivers. The true objective value is generally not available. $\tilde{n}_r$ indicates the subsampled number of receivers. At every iteration, the number of PDEs that need to be solved is $\tilde{n}_r + \tilde{n}_s$, which is much smaller than $n_r + n_s$.

Algorithm 3 describes the full algorithm, where the update direction is as in a gradient-descent type algorithm. The gradient update direction can also be replaced with other update directions, such as the Gauss-Newton update. A line-search is comparing current objective value estimates to previous objective value estimates. Because the data is changing every iteration, we found that a standard line-search (comparing $\tilde{f}(\mathbf{m}_k)$ to $\tilde{f}(\mathbf{m}_{k-1})$) causes the algorithm to stall after only a few iterations. To address this issue, we opt for a non-monotone line-search (see e.g., Birgin et al., 1999), which compares the current objective to the maximum of the previous few values (5 or 10 seem to be reasonable choices for seismic inverse problems). The algorithm described above is summarized in Algorithm 3.

Algorithm 3 Stochastic optimization algorithm to minimize $f(\mathbf{m}) = \sum_{i=1}^{n_s} f_i(\mathbf{m})$.

iteration counter $k = 1$, set sufficient descent parameter c
while not converged
1a. $\tilde{\mathbf{q}}$ `//draw 1 or a few source samples`
1b. $\tilde{\mathbf{p}}$ `//draw 1 or a few receiver samples`
`//approximate function value and gradient`
2. $\tilde{f}(\mathbf{m}) = \frac{1}{2}\sum_{i=1}^{\tilde{n}_s} \|\tilde{P}A(\mathbf{m}_k)^{-1}\tilde{\mathbf{q}}_i - \tilde{\mathbf{d}}_i\|_2^2$
3. $\tilde{\mathbf{g}} = \sum_{i=1}^{\tilde{n}_s} \mathbf{g}_i$
4a. $f_{\text{ref}} = \{\tilde{f}_k, \tilde{f}_{k-1}, \ldots, \tilde{f}_{k-M}\}$
4b. $\gamma = 1$
4c. **if** $\tilde{f}(\mathbf{m}_k - \gamma\tilde{\mathbf{g}}) < \max(f_{\text{ref}}) + c$
$\mathbf{m}_{k+1} = \mathbf{m}_k - \gamma\tilde{\mathbf{g}}$ `// update model estimate`
$k = k + 1$
else
$\gamma = \eta\gamma$ `//step size reduction,` $\eta < 1$
go back to 4c
end

Parallel adjoint-state

	Adjoint-state	Parallel reformulation
nr. of Helmholtz solves	2 per source	1 per source + 1 per receiver
Helmholtz solves per source	2 sequential	1+1 parallel
memory for fields per node	2 fields per source	1 source field or 1 receiver field

Table 1: Properties of different algorithms.

APPLYING THE (INVERSE) OF THE GAUSS-NEWTON HESSIAN WITHOUT PDE SOLVES

By construction, the Gauss-Newton Hessian is a symmetric positive (semi-)definite matrix, $H_{\text{GN}} \in \mathbb{C}^{N\times N}$, given by

$$H_{\text{GN}} = J^*J = \sum_{i=1}^{n_{\text{s}}} G(\mathbf{u}_i)^* A^{-*} P^* P A^{-1} G(\mathbf{u}_i).$$

The meaning of the partial derivative $G(\mathbf{u}_i) = \left(\frac{\partial A(\mathbf{m})\mathbf{u}_i}{\partial_{\mathbf{m}}}\right)$ is as described earlier in this abstract. The structure of the Gauss-Newton Hessian immediately shows that the maximum rank of H_{GN} is $\tilde{n}_{\text{r}} \times \tilde{n}_{\text{s}}$ for each frequency. In the stochastic optimization setting, where we work with a few sources and receivers at a time, it means that the Gauss-Newton Hessian is usually not full rank, because $\tilde{n}_{\text{r}} \times \tilde{n}_{\text{s}} \leq N$.

In this situation where the Hessian approximation is rank deficient, we can use a trust-region optimization method (Nocedal and Wright, 2000,chapter 4) or we can use a line-search algorithm in combination with some quadratic regularization.

A trust-region algorithm based on the ℓ_2-norm minimizes $f(\mathbf{m})$ and leads to subproblems of the form $\mathbf{p} = -(H_{\text{GN}} + \rho I_N)^{-1}\mathbf{g}$. The vector $\mathbf{p}$ is the update direction and ρ is a scalar related to the trust-region constraint in the equivalent subproblem $\min_{\mathbf{p}} f(\mathbf{m}) + \mathbf{p}^*\mathbf{g} + \frac{1}{2}\mathbf{p}^* H_{\text{GN}}\mathbf{p}$ s.t. $\|\mathbf{p}\|_2 \leq \delta$ where δ is the scalar trust-region radius, which is adapted every outer iteration.

When using a line-search algorithm, regularization can make H_{GN} an invertible and positive-definite matrix. We do this by minimizing the original objective with some quadratic regularization added: $f(\mathbf{m}) + \frac{\rho}{2}\|\mathbf{m}\|_2^2$ with the scalar $\rho > 0$. The corresponding Gauss-Newton Hessian equals $H_{\text{GN}} + \rho I_N$, the scaled identity term shifts the small eigenvalues away from zero to positive values. This regularization term does not need to be the identity but can be any positive-definite matrix, which gives us flexibility to include prior information on the model. With this additional term, the Gauss-Newton update direction is computed as $\mathbf{p} = -(H_{\text{GN}} + \rho I_N)^{-1}\mathbf{g}$.

Both Conjugate-Gradients (CG), LSQR or other iterative least-squares methods can be used to invert the regularized Gauss-Newton Hessian by either inverting $(H_{\text{GN}} + \rho I_N)$ or the system matrix $\begin{pmatrix} J \\ \sqrt(\rho) I_N \end{pmatrix}$. Both these methods only need matrix-vector products that can be written as a chain of operations—i.e. $\mathbf{y} = J\mathbf{x}$ can be computed as $\mathbf{y} = G(\mathbf{u}_i)\mathbf{x} \rightarrow \mathbf{y} = A^{-1}\mathbf{y} \rightarrow \mathbf{y} = P\mathbf{y}$ during which the PDEs are solved on the fly. When we use iterative methods, memory requirements are low but computational requirements are high since we need two PDEs solves per LSQR iteration.

By directly utilizing the fields $\mathbf{u}$ and $\mathbf{w}$, Habashy et al. (2011) compute the action of Gauss-Newton Hessians on vectors without solving PDEs. In this situation, the Hessian can be written as

$$H_{\text{GN}} = \sum_{i=1}^{n_{\text{s}}} G(\mathbf{u}_i)^* W W^* G(\mathbf{u}_i).$$

In this alternative formulation, the action of the Gauss-Newton Hessian can be computed PDE-free. In practice, this approach can be much faster than solving PDEs on-the-fly, provided the communication times of products with the distributed matrix W are small compared to the time it would take to solve the PDEs. In a parallel setting where W is column distributed, matrix-vector products of the form $W\mathbf{y}$ can be expensive in terms of communication, because it requires accessing elements of W across the dimension in which the data is distributed across nodes. Products with W^*, on the other hand, are much cheaper. Since the each column of W and each $\mathbf{u}_i$ is stored on a different compute node, application of H_{GN} to a vector depends on extensive communication for 3D problems, further investigation will be needed to determine if this approach scales. In this case, (in exact arithmetic) it would require at most $\tilde{n}_{\text{r}} \times \tilde{n}_{\text{s}} + 1$ CG iterations; 1 iteration per distinct eigenvalue. Each iteration needs one matrix-vector product with W and one with W^*. However, CG will require a smaller number of iterations if ρ is sufficiently large so it induces clustering of the eigenvalues, leading to a smaller number of distinct eigenvalues and therefor CG iterations.

To avoid significant parallel communication at every CG/LSQR iteration and shift the communication cost to a single computation, we can directly invert $H_{\text{GN}} + \rho I_N$ by applying the Sherman-Morrison-Woodbury identity. This results in the following expression for the inverse:

$$(H_{\text{GN}} + \rho I_N)^{-1} = \\ \frac{1}{\rho} I_N - \frac{1}{\rho} G(\mathbf{u})^* W (I_{\tilde{n}_{\text{r}}\tilde{n}_{\text{s}}} + W^* G(\mathbf{u}) \frac{1}{\rho} G(\mathbf{u})^* W)^{-1} W^* G(\mathbf{u}) \frac{1}{\rho}.$$

The only matrix that needs to be inverted now equals $(I_{\tilde{n}_{\text{r}}\tilde{n}_{\text{s}}} + W^* G(\mathbf{u}) \frac{1}{\rho} G(\mathbf{u})^* W) \in \mathbb{C}^{\tilde{n}_{\text{r}}\tilde{n}_{\text{s}} \times \tilde{n}_{\text{r}}\tilde{n}_{\text{s}}}$. The inverse of this very small matrix can always be computed explicitly. The full inverse Hessian is not formed, but applied as a chain of matrix-vector products. It is nontrivial to implement this communication efficient, given W is distributed over its columns. We leave evaluation of this option for future research.

NUMERICAL EXAMPLE

This frequency domain seismic waveform inversion example is based on algorithm 3 and uses only gradient and objective information. The true model is the 2D Marmousi model with a water layer of 200m on top. The sources are 50m apart and located near the water surface. The receivers are located on the ocean floor and there is also 50m distance between the receivers. A Ricker wavelet with 10Hz peak frequency was used. We employ a frequency continuation strategy, starting with 3Hz data, ending with 10Hz data. The true, initial and final models are shown in Figure 1 . The result using 8 sources and 8 receivers is a bit noisy, but the larger scale characteristics are at the correct locations. There is little difference between the result with all sources and all receivers active at every iteration, and the one where 16 sources and receivers are used. The results using 8 and 16 sources/receivers take the same time to compute, but the number of compute nodes is $8+8$ versus $16+16$. The result using all sources and receivers takes twice as long to compute, because it is based on the sequential adjoint state algorithm. Algorithm 2 cannot be used in this case, as it would require an excessive number of compute nodes.

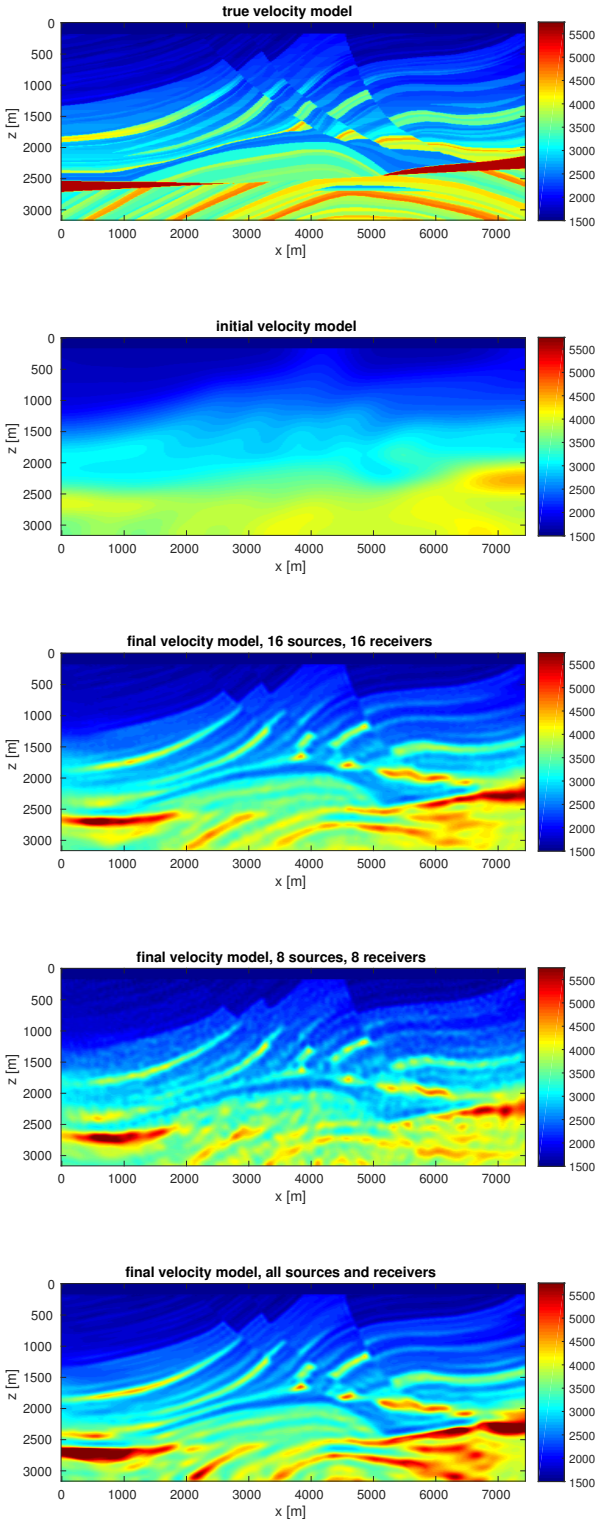

Figure 1: True, inital and estimated models for various subsampling ratios.

CONCLUSIONS

In this work we presented a parallel reformulation of the sequential adjoint-state method to compute the gradient of a nonlinear least-squares objective function. The reformulation enables parallel solution of the forward and adjoint PDE. The proposed algorithm requires one PDE solve per source and one per receiver. This results in a $2\times$ speedup if the number of sources plus the number of receivers is equal to the number of PDEs that can be solved in parallel on the available computational resources. The computations are the same as the adjoint-state method requires: solutions of linear systems involving the PDE, $A(\mathbf{m})$, and its adjoint $A(\mathbf{m})^*$. A byproduct of the parallelized adjoint-state algorithm is the availability of fields which can be used to apply the Gauss-Newton Hessian without the need to recompute wavefields. A practical stochastic optimization strategy with source/receiver randomization and subsampling is used to show good waveform inversion results can be obtained with only a few active sources and receivers at every full-waveform inversion iteration. Future work will focus on large scale 3D waveform inversion problems. In 3D, the main question is how many sources and receivers are required at each FWI iteration.

ACKNOWLEDGEMENTS

This work was financially supported in part by the Natural Sciences and Engineering Research Council of Canada Collaborative Research and Development Grant DNOISE II (CDRP J 375142-08). This research was carried out as part of the SINBAD II project with the support of the member organizations of the SINBAD Consortium.

REFERENCES

Birgin, E. G., J. M. Martínez, and M. Raydan, 1999, Nonmonotone spectral projected gradient methods on convex sets: SIAM J. on Optimization, **10**, 1196–1211.

Habashy, T. M., A. Abubakar, G. Pan, and A. Belani, 2011, Source-receiver compression scheme for full-waveform seismic inversion: GEOPHYSICS, **76**, R95–R108.

Haber, E., M. Chung, and F. Herrmann, 2012, An Effective Method for Parameter Estimation with PDE Constraints with Multiple Right-Hand Sides: SIAM Journal on Optimization, **22**, 739–757.

Krebs, J. R., J. E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M.-D. Lacasse, 2009, Fast full-wavefield seismic inversion using encoded sources: GEOPHYSICS, **74**, WCC177–WCC188.

Nocedal, J., and S. J. Wright, 2000, Numerical optimization: Springer.

Symes, W., 2010, Source synthesis for waveform inversion: Presented at the 2010 SEG Annual Meeting, Society of Exploration Geophysicists.

van Leeuwen, T., A. Y. Aravkin, and F. J. Herrmann, 2011, Seismic waveform inversion by stochastic optimization: International Journal of Geophysics, **2011**.

van Leeuwen, T., and F. J. Herrmann, 2013, Fast waveform inversion without source-encoding: Geophysical Prospecting, **61**, 10–19.