SIAM J. OPTIM.
Vol. 22, No. 3, pp. 739–757

AN EFFECTIVE METHOD FOR PARAMETER ESTIMATION WITH PDE CONSTRAINTS WITH MULTIPLE RIGHT-HAND SIDES*

ELDAD HABER†, MATTHIAS CHUNG‡, AND FELIX HERRMANN§

Abstract. Often, parameter estimation problems of parameter-dependent PDEs involve multiple right-hand sides. The computational cost and memory requirements of such problems increase linearly with the number of right-hand sides. For many applications this is the main bottleneck of the computation. In this paper we show that problems with multiple right-hand sides can be reformulated as stochastic programming problems by combining the right-hand sides into a few "simultaneous" sources. This effectively reduces the cost of the forward problem and results in problems that are much cheaper to solve. We discuss two solution methodologies: namely sample average approximation and stochastic approximation. To illustrate the effectiveness of our approach we present two model problems, direct current resistivity and seismic tomography.

Key words. PDE constrained optimization, stochastic optimization, stochastic approximation, sample average approximation, inverse problems, parameter estimation, large scale problems, DC resistivity, seismic tomography, multiple right hand sides

AMS subject classifications. 35Q93, 35R30, 62L20, 65M32, 65N21, 90C15

DOI. 10.1137/11081126X

1. Introduction.

1.1. Background.

In this work we consider the solution of discretized parameter estimation problems with PDEs as constraints. We start by restricting ourselves to a particular form,

$$\min_{y_1,\dots,y_N,u} J(y_1,\dots,y_N,u) = \frac{1}{2N}\sum_j \|Py_j - d_j\|^2 + R(u), \tag{1.1a}$$

$$\text{s.t.} c_j(y_j,u) := A(u)y_j - q_j = 0, \qquad j = 1,\dots,N. \tag{1.1b}$$

Here, $u \in \mathbb{R}^m$ is the discretized model (or control) and $y_j \in \mathbb{R}^n$, $j = 1,\dots,N$, are the discretized fields (or states). Without loss of generality, we assume the underlying PDE to be time invariant, but our method lends itself readily to be extended to the time-varying case. The constraints $c_j(y_j,u)$ are discretized linear PDEs. The matrix $A(u)$ is a discretization of the PDE (system matrix) that depends on the model u, and $y_1,\dots,y_N$ are the fields that correspond to the sources, $q_1,\dots,q_N$.

To keep the discussion succinct, we follow a discretize-then-optimize approach, where A is assumed to be invertible for all relevant u. An analogous formulation in the optimize-then-discretize framework is also possible but will not be discussed here. The objective function J is composed of two terms. The first term represents the data-misfit function, defined as the energy difference between the solution of the PDE

*Received by the editors October 11, 2010; accepted for publication (in revised form) March 26, 2012; published electronically July 3, 2012.
http://www.siam.org/journals/siopt/22-3/81126.html

†Departments of Mathematics and Earth and Ocean Science, University of British Columbia, Vancouver, BC, Canada (haber@math.ubc.ca). The work of this author was funded by the NSERC IRC program.

‡Department of Mathematics, Virginia Tech, Blacksburg, VA 24061 (mcchung@vt.edu). The work of this author was supported by the Texas State University Research Enhancement Program.

§Department of Earth and Ocean Science, University of British Columbia, Vancouver V6T 1Z4, BC, Canada (felix@eos.ubc.ca).

y_j projected by the matrix P to the data vector d_j associated with the jth source. The second term, $R(u)$, is a convex regularization functional. PDE-constrained optimization problems of this type arise in parameter recovery in impedance tomography, DC resistivity, electromagnetic imaging, seismic inversion, hydrology, and more [20, 26, 7, 14]. However, notice that the above formulation is a special case because the acquisition geometry, i.e., the locations of the receivers encoded in the rows of P, remains invariant among the N different source experiments. In realistic settings such as in marine acquisition, where sources and receivers both move, this simplification forms a serious restriction that we address later in this paper.

In the last few years many algorithms have been developed for parameter estimation with PDE constraints; see, for example, [5, 11, 17] and reference within. It is straightforward to show that the necessary conditions for the solution of the problem is obtained by solving the nonlinear systems of equations

$$A(u)y_j - q_j = 0, \quad j = 1, \dots, N, \tag{1.2a}$$

$$A(u)^\top \lambda_j + \frac{1}{N} P^\top (Py_j - d_j) = 0, \quad j = 1, \dots, N, \tag{1.2b}$$

$$\nabla_u R(u) + \frac{1}{N} \sum_j G(y_j, u)^\top \lambda_j = 0, \tag{1.2c}$$

where

$$G(y_j, u) = \nabla_u [A(u)y_j] \tag{1.3}$$

is the gradient and the variables $\lambda_j,\ j = 1, \dots, N$, are a set of Lagrange multipliers. Since the problem is nonconvex, an appropriate strategy is needed to guarantee convergence to a local minima (see [9] and [24, Ch. 18]).

Unfortunately, in many applications the size of the field y_j and the number of right-hand sides N are too large to use an "all at once" approach during which these three equations are solved jointly. The storage requirement alone would be at least $2N \times n + m$, which for realistic three-dimensional problems is prohibitive given the large grid size and the large number of required sources. By employing the so-called reduced space method—which requires only $2n + m$ storage while being less sensitive to the number of PDEs—we can address this storage issue.

However, this approach requires a parameter-to-field map $y_j = y_j(u) = A(u)^{-1} q_j$, which involves inversion of the discretized PDE. After elimination of the PDE constraint in (1.1), the resulting unconstrained optimization problem becomes

$$\min_u \quad J(u) = \frac{1}{2N} \sum_j \|PA(u)^{-1} q_j - d_j\|^2 \ + \ R(u). \tag{1.4}$$

It is straightforward to show [24, 13] that the solution of the problem satisfies the following (reduced gradient) nonlinear equation:

$$\nabla_u J(u) = \frac{1}{N} \sum_j G(y_j, u)^\top A(u)^{-\top} P^\top (PA(u)^{-1} q_j - d_j) \ + \ \nabla_u R(u) = 0. \tag{1.5}$$

Note that in case of nonconvex problems one may need to consider globalization methods for optimization [24].

Examining this expression, one can observe that each source requires N PDE solves for the forward and adjoint systems. This cost is a major impediment because modern-day parametric inversions have to deal with surveys that have increasingly large numbers of sources.

1.2. Motivation. Solutions of equations of the type in (1.5) require an iterative scheme that involves several steps. Algorithm 1 contains an example of such a representative iterative scheme, where the solution to (1.5) is found by an iterative descent algorithm with a large "for loop" (lines 4–6) that consists of three steps: the solution of the forward (line 5), the adjoint (line 6) systems involving the solution of $2N$ PDEs, and the computation of the gradient update (line 7).

This for-loop can be trivially parallelized. Nonetheless, if N is very large, then this loop is the main bottleneck of the computation. In fact, the overall complexity of the algorithm can be roughly estimated by the number of PDE solves, which is $\mathsf{iter} \times 2\,N$, where iter is the number of iterations of the external optimization problem. Indeed, even if the number of iterations, iter, is small, for large N the computational burden is large. The question at hand is, how can we reduce the computational cost of solving the optimization problem?

ALGORITHM 1. Computation of an iteration.

1: Initialize the solution $u = u_0$
2: **while** not converge **do**
3: Initialize the gradient $g_k = \nabla R(u^{k-1})$
4: **for** $j = 1, \ldots, N$ **do**
5: Solve $A(u^{k-1})y_j^k = q_j$
6: Solve the linear system $A(u^{k-1})^\top \lambda_j^k = -\frac{1}{N} P^\top (P y_j^k - d_j)$
7: Set $g_k \leftarrow g_k + G(y_j^k, u^{k-1})^\top \lambda_j^k$
8: **end for**
9: set $u^k = u^{k-1} - \gamma H_k^{-1} g_k$ where H_k is an approximation to the Hessian and γ is chosen by a "soft" line search.
10: **end while**

1.3. Existing work. Three approaches have been proposed to tackle such problems. Here, we summarize them briefly:

- If the size of the linear system $A(u)$ is not too large, then it is possible to use direct factorization to deal with multiple right-hand sides. This was the approach taken in [26], but it is restricted to problems where the filling of the Cholesky factors is sufficiently small and computer memory is sufficiently large.
- Iterative solutions to the linear systems have been used for large-scale problems. To avoid the overall cost, it has been suggested to recycle the Krylov vectors [18]. This approach saves some computation but still requires a large amount of time when the number of sources is large.
- A third approach, which originated in seismic data processing and is the most economical approach, is referred to as simultaneous random sources. In this method we combine right-hand sides and data [4, 28, 15, 22, 27]. Taking linear combinations of sources one can obtain a "new" source

$$\widehat{q} = \sum_j w_j q_j$$

and a "new" data

$$\widehat{d} = \sum_j w_j d_j,$$

where w_j are weights to be chosen by some criteria. One then replaces the optimization problem (1.4) with

$$\min_u \quad J(u) = \frac{1}{2}\|PA(u)^{-1}\widehat{q} - \widehat{d}\|^2 \; + \; R(u), \tag{1.6}$$

which requires only a single solution of the PDE that can be done iteratively. It seems rather surprising but in some cases this approach yields results that are not far from using the complete data set. Nonetheless, while the approach has been successful it is highly "add-hocish." Choosing the weights, the quality of the solution and the comparison of the solution with the solution of the original problem are not fully understood. In some cases, the weights are chosen based on arguments such as wave interference, source focusing, and noise whitening. As we show here, these arguments can lead to suboptimal performance.

In the following we propose a methodology that allows us to use a similar approach to (1.6) of reducing the problem with multiple sources to a problem with a single or a few sources. This allows us to reduce the computational cost of the problem from $2\,\mathsf{iter} \times N$ to roughly $2\,\mathsf{iter} \times r$ with $r \ll N$. We show that the minimization problem (1.6) is a particular (rather incomplete) instance of our approach. We also give a firm theoretical justification to this formulation and show how it can be solved efficiently.

Our approach shares many similarities to deterministic problems in machine learning, where large amounts of data are treated in a stochastic manor (see, e.g., [6] and reference within) in order to reduce the computational cost. Similar ideas have been proposed in the context of the solutions of least squares problems (see [2] and reference within), where randomized algorithms are used to solve deterministic problems.

The following two examples are realistic model problems that we use to demonstrate the concepts in the paper.

Example 1 (the DC resistivity problem [25]). Consider the case where the matrix A results from the discretization of the problem

$$\nabla \cdot \; \exp(u)\,\nabla y_j = q_j, \qquad \nabla y_j \cdot \vec{n} = 0, \qquad \int_\Omega y_j = 0 \qquad j = 1, \ldots, N.$$

In this case the goal is to recover the log conductivity u given measurements of the potential fields y_j which results from the sources q_j. Modern acquisition systems can have hundreds and thousands of sources spread over a physical grid and in boreholes. Upon a finite volume or a finite element discretization of the differential equation one obtains the PDE (1.1b).

Example 2 (the full-waveform inversion problem [1]). Consider the case where the matrix A results from the discretization of the problem

$$\left(\Delta + k^2(u)\right) y_j = q_j \;\; \text{in } \Omega \subset \mathbb{R}^2, \;\; y_j = 0 \text{ on } \Gamma = \partial\Omega, \;\; j = 1, \ldots, N,$$

where $k^2(u) = 2\pi f u$ is the wave number with f the temporal frequency, and $u = \mathrm{c}^{-2}$ with c the acoustic wave-speed. In this case the goal is to recover the velocity $\mathrm{c} = u^{-1}$ given measurements of the wave-field fields y_j for each source experiment.

1.4. This paper. Our aim is to put empirical findings on fast parameter estimation [19] on a more rigorous footing using the theory of stochastic programming [31, 23]. This allows us to obtain better algorithms for the problem as well as to extend the algorithm to problems where previous techniques do not succeed.

The paper is structured as follows. In section 2 we present the bulk of the paper and introduce the basic concept of our reformulation. We show that the problem can be reformulated as a stochastic programming problem. We then discuss two commonly used techniques to solve this type of problem. In section 3 we discuss the quality of the approximation compared to the original problem. In section 4 we extend the method to problems where sources may have different receivers. In section 5 we explore different algorithms, and finally, in section 6 we summarize the paper and make some concluding remarks.

2. A stochastic perspective of the optimization problem. In this section, we recast (1.1) as a stochastic programming problem involving estimation of the trace. We also introduce two different stochastic approximation techniques that we will use to receive a solution to the optimization problem (1.1).

2.1. A stochastic programming interpretation. In order to derive a fast algorithm for the optimization problem we require a different point of view that allows us to explore new algorithmic horizons. We do that by considering the problem as a stochastic programming problem. For simplicity we consider only the unconstrained formulation and rewrite it first as

$$\min_u \; J(u) = \frac{1}{2}\|PA(u)^{-1}Q - D\|_F^2 \; + \; R(u), \tag{2.1}$$

where $Q = (q_1, \ldots, q_N)$, $D = (d_1, \ldots, d_N)$, and $\|\cdot\|_F$ denotes the Frobenius norm. The difficulty, of course, is to evaluate the residual term $S(u) = PA(u)^{-1}Q - D$ and its norm. We therefore turn now to a stochastic interpretation of the trace. Recall that for a random vector w with 0 mean and identity covariance matrix I we have that

$$\mathbf{E}_w \left(w^\top S(u)^\top S(u) w\right) = \mathsf{trace}\left(S(u)^\top S(u)\right) = \|S(u)\|_F^2, \tag{2.2}$$

where $\mathbf{E}_w$ is the expected value with respect to w that belongs to a probability space $W : \Omega \to \mathbb{R}^n$. Using this identity we replace the deterministic problem (2.1) with a stochastic programming problem

$$\min_u J(u) := \mathbf{E}_w \left(J_W(u, w)\right), \tag{2.3}$$

$$\text{where } J_W(u, w) \; = \; \frac{1}{2}\left(\|(PA(u)^{-1}Q - D)w\|^2\right) + \; R(u).$$

By virtue of the identity (2.2), the problem (2.3) is identical to problem (2.1). Because the latter has the form of a stochastic programming problem we gain access to stochastic programming methods. As long as the evaluation of the expectation in (2.3) can be done efficiently and with sufficient accuracy, we gain in efficiency. Remember that in the stochastic formulation, each realization w_j of w involves a single solve for the discrete PDE and

$$(PA(u)^{-1}Q - D)w_j = PA(u)^{-1}\left(Qw_j\right) - Dw_j$$

implies that for a given w_j only a *single* PDE is solved in order to evaluate the stochastic misfit function. This is in contrast to the N PDEs to be solved when evaluating (2.1). This observation is used to obtain very efficient algorithms.

In particular, two end members of stochastic programming algorithms can be considered.

- In the sample average approximation (SAA) technique one approximates the expected value by a Monte Carlo approximation using the sum

$$\mathbf{E}_w\left(J_W(u,w)\right) \approx \frac{1}{K}\sum_{j=1}^{K} J_W(u,w_j),$$

  where w_j are samples chosen from an appropriate probability density function (see the discussion in the next subsection). The idea is to use $K \ll N$ different realizations to approximate the expected value.
- A second class of algorithms for the solution of stochastic programming problems is referred to as stochastic approximation (SA) techniques. In these techniques one uses a *single* realization and computes the gradient $\nabla_u J(u,w_j)$. A step in the negative direction of the gradient is then taken, averaged with the previous steps, and the process repeats. The point here is that for each step of the SA method only a *single* realization is needed. For our problem this implies that we require a single PDE and adjoint to be solved at each iteration (rather than $2N$ in the original problem and $2K$ in the SAA approach).

Both approaches can lead to increased performance because each approximation may require fewer PDE solves. Before we can discuss the performance of these approximations, let us first formulate these two approximations in further detail for our parameter estimation problem.

2.2. Solution through the SAA. Maybe the simplest way to solve the stochastic problem is by using the SAA [31]. Here we replace the expectation in the minimization problem (2.3) by the approximation

$$\mathbf{E}_w(J_W(u,w)) \approx \widehat{J}_K(u;w_1,\ldots,w_K) := \frac{1}{2K}\sum_{j=1}^{K} \|PA(u)^{-1}Qw_j - Dw_j\|^2 + R(u). \tag{2.4}$$

Note that unless $K \ll N$ we are no better off than solving the original problem, as the number of PDE solves is $2K$ per iteration. An interesting observation was made in [3], where different methods were experimented with to approximate the trace of matrices. It has been observed that in many cases, even for $K = 1$ the evaluation of the trace was sufficiently accurate. For instance, in [10] a single evaluation was used to minimize the general cross validation function with satisfactory results.

In order to choose the vectors w_j we turn to stochastic trace estimators [16]. We have seen that if w has 0 mean and identity covariance matrix I, then $\mathbf{E}_w(w^\top S^\top S w) = \mathsf{trace}(S^\top S)$. It is interesting to observe that the expected value is identical for all w's drawn from *any* distribution which is 0 mean and covariance I. However, since we are heading for approximation, our goal is to obtain this mean with the smallest variance possible; this can help in generating the smallest amount of right-hand sides in the PDE. It has been shown in [16] that this can be obtained when w is a random vector with independently distributed components each taking values 1 or -1 with equal probability $1/2$, also known as a Rademacher distribution. We therefore choose w from this distribution. Note that although other choices of w are possible, they are suboptimal, in the sense that their variance is higher.

The advantage of the SAA technique is that it separates the questions of approximating the expectation and the optimization algorithm to be used. In order to solve the optimization problem any robust algorithm can be used. Here for simplicity and for comparison reasons we have used the LBFGS method applied to the uncon-

strained problem [24, 13]. Other (and in our opinion better) choices can be made for the solution of the problem. We comment on this choice later in this work.

2.3. Solution through SA algorithms. We now discuss a family of algorithms that enable us to obtain fast solutions for optimization problem (2.3). The general structure of an SA algorithm is as summarized in Algorithm 2. The algorithm is a simple modification of a steepest descent algorithm with constant step size. Theory for other (less primitive) methods such as Gauss–Newton and limited memory BFGS do not exist (as far as we know). However, applying these methods successfully have been reported in machine learning applications [30]. We have experimented with both theoretically sound techniques and techniques that do not have convergence proofs but seem to give good results in practice.

ALGORITHM 2. SA for inverse problem.

1: Initialize the solution $u = u_0$
2: **while** not converge **do**
3: choose a realization w_k
4: set $s = -\nabla J_W(u_k, w_k)$
5: set $\widehat{u}_{k+1} = u_k + \gamma s$
6: average $u_{k+1} = \frac{1}{k+1}\left(\sum_j u_j + \widehat{u}_{k+1}\right)$
7: **end while**

When implementing the algorithm, there are two questions at hand: how to pick an appropriate realization w_k and how to choose the step size γ in step 5 of the algorithm. We have already seen that an optimal choice of w is obtained by choosing w to be a realization from a Rademacher distribution. The choice of the step size is more delicate. Although there is a well established theory for its selection we have found that it can be difficult to practically choose a step size that works well. Some robust versions exist [23] but they require convexity as well as a bounded domain, which are not natural for our problems. Establishing a convergence theory for stochastic programming methods for the nonconvex PDE constraint optimization problem of this kind is rather difficult. However, many investigations of these nonconvex problems on various applications show promising results and the method used despite the lack of a sound convergence theory [32, 4, 28, 15, 22, 27]. We are aware of the fact that an appropriate convergence theory is desirable.

3. The quality of the approximation. The quality of our approximation heavily depends on the stochastic approximation of (2.2). For statistical problems, convergence is guarantied as the number of samples go to infinity. Since we use Monte Carlo methods, convergence is typically related to the square root of the number of samples. The error is therefore associated with the fact that the problem is approximated using a finite number of sampling points and thus the variance of the solution is the relevant quantity to look at.

It is important to note that the slow rate of convergence in the number of samples implies that recasting deterministic problems as stochastic programming problems is appropriate only for problems where the required accuracy is relatively low. Inverse problems are such a typical example, as the objective function depends on unknown parameters (such as regularization parameter), and therefore, as we see next, the solution obtained by the reformulation is adequate.

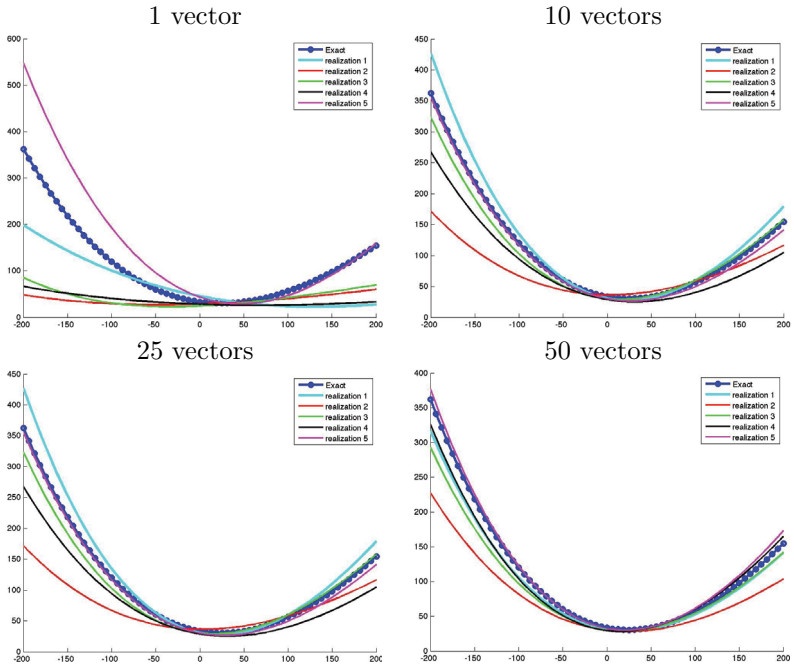

FIG. 1. *Approximation to the misfit in one direction using a single vector (top left),* 10 *vectors (top right),* 25 *vectors (bottom left), and* 50 *vectors (bottom right). For each approximation we have used* 5 *different realizations.*

3.1. An illustrative example. To motivate and illustrate a discussion we use Example 1, namely, the DC resistivity example presented in the introduction. As we have seen, the goal is to approximate the misfit

$$\mathsf{misfit}(u) = \|PA(u)^{-1}Q - D\|_F^2$$

using SA. To have a feel for the approximation we set u to be constant, e.g., 10^{-2}, compute the gradient of the misfit, $s = \nabla_u \mathsf{misfit}(u)$, and then define a one-dimensional function

$$f(\alpha) := \mathsf{misfit}(u + \alpha s) = \|PA(u + \alpha s)^{-1}Q - D\|_F^2.$$

We then use an approximation

$$\widehat{f}_K(\alpha) = \widehat{\mathsf{misfit}}(u + \alpha s; w_1, \ldots, w_K) = \frac{1}{K}\sum_{j=1}^{K} \|(PA(u + \alpha s)^{-1}Q - D)w_j\|^2.$$

For our experiments we have 500 sources and we discretize the PDE using 32^3 cells, where y and u are discretized on cell-centers [12]. Thus $Q : \mathbb{R}^{500} \to \mathbb{R}^{32^3}$ and to compute the misfit in the original formulation $A(u)$ has to be inverted 500 times. We then use random realizations $[w_1, \ldots, w_K]$, where we experiment with $K = 1, 10, 25, 50$. Evaluating the misfit in this way is roughly a factor of $500K^{-1}$ cheaper compared to the original evaluation. In Figure 1 we plot the estimated misfit $\widehat{\mathsf{misfit}}$ as a function of α. Since each approximation depends on a particular choice of w, that is, random, we

use five different realizations of w for each approximation. We see that even though the functions are not identical, using a single vector to evaluate the misfit, the minimum with respect to α is not far from the minimum obtained by exact evaluation of the misfit. This shows why methods that reduce the data to data that is collected from a single source work well in practice.

3.2. A quantitative evaluation. The above discussion motivates the use of very few samples in the SAA. However, it is often observed in practice that the bounds for the number of samples needed for the approximation is rather pessimistic [31]. Better results can be obtained by direct evaluation of the quality of the solution. Here, we review and adapt a simple methodology discussed in [31] for the estimation of the quality of the solution obtained by the SAA method.

We will use the following notation. By $w_{[1,K]}$ we mean a sample of K vectors $[w_1, \ldots, w_K]$. By $w^{\ell}_{[1,K]}$ we mean the ℓth sample of K vectors. Let $u^\star$ be the minimizer of the approximate objective function. Thus, for a given $w_{[1,K]}$ we have that

$$u^\star = \arg\min_u \widehat{J}_K(u; w_{[1,K]}) = \frac{1}{K}\sum_j J(u; w_j).$$

Let $u^\dagger$ be the minimum of the exact objective function;

$$u^\dagger = \arg\min_u J(u) = \mathbf{E}_w(J_W(u, w)),$$

hence

$$J(u^\dagger) \leq J(u^\star).$$

The quality of the approximation can be measured by the gap

$$\mathsf{gap} = J(u^\star) - J(u^\dagger). \tag{3.1}$$

Obviously, this gap cannot be computed exactly. However, it can be estimated statistically.

First, we would like to estimate $J(u^\star)$. We thus resample $w_{[1,K]}$ and compute the value of $\widehat{J}_K(u^\star; w_{[1,K]})$ for different realizations of $w_{[1,K]}$. If we sample $w_{[1,K]}$ L times, then we can estimate the mean and variance obtaining

$$J(u^\star) \approx \frac{1}{L}\sum_\ell \widehat{J}_K(u^\star; w^\ell_{[1,K]}) = J^\star_L.$$

Notice that in order to compute this estimate we do not require solving an optimization problem. At this step we fix $u = u^\star$ and change the samples $w_{[1,K]}$. Given the realizations we can also estimate the variance of $\widehat{J}_K(u^\star; w_{[1,K]})$

$$(\sigma^\star)^2 = \frac{1}{L(L-1)}\sum_\ell (\widehat{J}_K(u^\star; w^\ell_{[1,K]}) - J^\star_L)^2.$$

The combination of the mean and the variance gives us a lower and an upper estimate on $J(u^\star)$. In particular, the 68% confidence interval is given by

$$J^\star_L - \sigma^\star \leq J(u^\star) \leq J^\star_L + \sigma^\star. \tag{3.2}$$

Evaluating $J(u^\dagger)$ requires more elaborate computational work. Consider as before independent and identically distributed samples $w^\ell_{[1,K]}$, $\ell = 1, \ldots, T$, and solve T optimization problems

$$u^\star_\ell = \arg\min_u \widehat{J}_K(u; w^\ell_{[1,K]}), \quad \ell = 1, \ldots, T.$$

We obtain T different solutions $[u^\star_1, \ldots, u^\star_T]$. Again, rather than obtaining the exact value of the function $J(u^\dagger)$ we evaluate its variance. Using the solutions $[u^\star_1, \ldots, u^\star_T]$ we can estimate the mean and variance of $J^\dagger$. Let

$$J^\dagger_T = \frac{1}{T} \sum_j \widehat{J}_K(u^\star_j; w^j_{[1,K]})$$

be the mean of the function value at the different solutions and let

$$(\sigma^\dagger)^2 = \frac{1}{T(T-1)} \sum_j (\widehat{J}_K(u^\star_j; w^j_{[1,K]}) - J^\dagger_T)^2$$

be the variance of the function of the different solutions. We then obtain that the estimated 68% confidence interval (that is, 68% of the solutions lie in this interval) is given by

$$J^\dagger_T - \sigma^\dagger \le J(u^\dagger) \le J^\dagger_T + \sigma^\dagger. \tag{3.3}$$

Combining the upper bound from (3.2) and the lower bound from (3.3) we can obtain an estimate on the quality of the solution. Although the lower bound is biased (since it involves solving an optimization problem), the bias is downward and therefore the estimate is useful. Note that unlike evaluating $J(u^\star)$, where only function evaluation are required, the evaluation of $J(u^\dagger)$ requires the solution of a few optimization problems. However, two points should be stressed here. First, in most cases, very few optimization problems need to be solved (say, 5 to 10; see a more elaborate discussion in [31] and [21]), and second, from a computational point of view solving these optimization problems can be done quickly if we start from the solution $u^\star$ of the first optimization problem. Furthermore, in the case that one has a parallel system at hand, these optimization problems are independent and can be solved in parallel.

4. An extension to the case of different observation operators. The algorithms developed so far heavily depend on the fact that different experiments represented by q_j have the same observation matrix P. Although many problems share this property, this is not always the case and we therefore want to extend the technique to a more general class of problems. To do that we rewrite the misfit as

$$\mathsf{misfit}(u) = \|C \odot (PA(u)^{-1}Q - D)\|_F^2 = \|C \odot S(u)\|_F^2, \tag{4.1}$$

where as before

$$S(u) = PA(u)^{-1}Q - D$$

is the residual matrix. Here $\odot$ is the pointwise Hadamard product. The matrix C is, in general, a dense matrix. To see that this is indeed a desired generalization we consider a few cases:

- If $C = \mathbf{1}$, that is, the matrix C is populated by 1's, then we are back to the previous case where all sources share all receivers.
- Consider the case that C is dense and populated by different positive numbers. This case typically arises where each datum has a different standard deviation and C_{ij} is the inverse of the squared standard deviation of each datum.
- If C is sparse, then we choose only a subset of source-receiver configuration.
- In the extreme case C is diagonal, which implies that each source has a single receiver. This configuration is common in many applied geophysical surveys where the source and receiver move simultaneously.

Just as in the previous sections we could represent the misfit by the expectation

$$\begin{aligned}\mathsf{misfit}(u) &= \|C \odot S(u)\|_F^2 \\ &= \mathbf{E}_w \left(\|(C \odot S(u))w\|^2 \right) = \mathbf{E}_w \left(\| \left(C \odot (PA(u)^{-1}Q - D) \right) w\|^2 \right).\end{aligned} \tag{4.2}$$

To understand the difficulty in the evaluation of the above expression recall that in the case where $C = \mathbf{1}$ we have

$$(\mathbf{1} \odot S(u))\, w = S(u)w = PA(u)^{-1}(Qw) - Dw.$$

However, for $C \neq \mathbf{1}$ one cannot simply follow the same methodology. The difficulty is that the Hadamard product does not associate with the matrix-vector product and therefore one cannot compute the product $S(u)w$ first and only then compute the Hadamard product. This is the main challenge in extending the method discussed in the previous sections.

There are a number of different ways to obtain a stochastic approximation to the trace in this case. Here, we explore two such approximations. The one we have found most useful is based on direct estimation of the product $C \odot S(u)$. It is straightforward to observe that if w is a random variable with 0 mean and identity covariance, then

$$C \odot S(u) = \mathbf{E}_w \left(\mathsf{diag}\left(S(u)w\right) C\, \mathsf{diag}\left(w\right) \right) \tag{4.3}$$

and therefore one arrives at the stochastic representation of the misfit

$$\mathsf{misfit}(u) = \|C \odot S(u)\|_F^2 = \|\mathbf{E}_w \left(\mathsf{diag}\left(S(u)w\right) C\, \mathsf{diag}\left(w\right) \right)\|_F^2 \tag{4.4}$$

and its approximation

$$\widehat{\mathsf{misfit}}(u) = \left\| \frac{1}{K} \sum_{j=1}^{K} \mathsf{diag}\left(S(u)w_j\right) C\, \mathsf{diag}\left(w_j\right) \right\|_F^2. \tag{4.5}$$

There is a fundamental difference between the approximation (4.5) and the ones we have discussed in the previous sections. In this case the expected value is *inside* the norm; hence we first require evaluating the expectation (approximated by the sum) inside and only then evaluate the norm. Notice that once again, only matrix vector product $S(u)w_j$ is needed in order to evaluate the misfit. Thus the number of PDEs to be solved is equivalent to the number of stochastic realizations.

To obtain the derivative of the approximation to the misfit we rewrite

$$\mathsf{vec}\left(\mathsf{diag}\left(S(u)w_j\right) C\, \mathsf{diag}\left(w_j\right)\right) = C_{w_j} S(u) w_j,$$

where vec refers to reordering the matrix as a long vector (column after column) and the matrix C_{w_j} is defined by

$$C_{w_j} = \left[\mathsf{blockdiag}((C\mathsf{diag}\left(w_j\right))_{1\rightarrow}, (C\mathsf{diag}\left(w_j\right))_{2\rightarrow}, \ldots, (C\mathsf{diag}\left(w_j\right))_{K\rightarrow})\right]^\top,$$

where $\ell \rightarrow$ implies the ℓth row of the matrix. Using this notation the approximated misfit can be rewritten as

$$\widehat{\mathsf{misfit}}(u) = \left\| \frac{1}{K} \sum_{j=1}^{K} C_{w_j} S(u) w_j) \right\|^2 = \left\| \frac{1}{K} \sum_{j=1}^{K} C_{w_j} (PA(u)^{-1} Q w_j - D w_j) \right\|^2 .$$

This representation allows for simple differentiation of the approximated misfit

$$\nabla_u \widehat{\mathsf{misfit}}(u) = - \left(\frac{1}{K} \sum_{j=1}^{K} G_j^\top A(u)^{-\top} P^\top C_{w_j}^\top \right) \left(\frac{1}{K} \sum_{j=1}^{K} C_{w_j} S(u) w_j) \right),$$

where the matrix $G_j = G(Qw_j, u)$ is defined by (1.3). This formulation of the problem requires K solves of the adjoint problem.

Other stochastic approximations to the trace can be obtained. Starting from (4.2) we first use the identity

$$(C \odot S(u))\, w = \mathsf{diag}\left(C\, \mathsf{diag}\,(w)\, S(u)^\top\right)$$

and therefore

$$\mathsf{misfit}(u) = \|C \odot S(u)\|_F^2 = \mathbf{E}_w \left(\|\mathsf{diag}\left(C\, \mathsf{diag}\,(w)\, S(u)^\top\right)\|^2 \right). \tag{4.6}$$

The identity (4.6) still does not solve the problem of computing either $A(u)^{-1}Q$ or $A(u)^{-\top}P$. The point here is that evaluating the diagonal of the matrix product

$$C\, \mathsf{diag}\,(w)\, S(u)^\top = C\, \mathsf{diag}\,(w)\, (PA(u)^{-1}Q - D)^\top$$

is needed *without* computing the matrix itself. We thus turn to yet another stochastic process. It can be shown [29] that if x is a vector with zero mean and covariance matrix $\mathsf{Cov}(x) = I$, then

$$\mathsf{diag}\left(C\, \mathsf{diag}\,(w)\, S(u)^\top\right) = \mathbf{E}_x \left(x \odot (C\, \mathsf{diag}\,(w)\, S(u)^\top) x \right). \tag{4.7}$$

Thus we replace the misfit with a double stochastic process

$$\mathsf{misfit}(u) = \|C \odot S(u)\|_F^2 = \mathbf{E}_w \left(\|\mathbf{E}_{x|w} \left(x \odot (C\, \mathsf{diag}\,(w)\, S(u)^\top) x \right) \|^2 \right). \tag{4.8}$$

Once again, we can obtain an approximation using Monte Carlo

$$\widehat{\mathsf{misfit}}(u) = \frac{1}{K_1 K_2} \sum_{i=1}^{K_1} \left\| \sum_{j=1}^{K_2} \left(x_j \odot (C\, \mathsf{diag}\,(w_i)\, S(u)^\top x_j \right) \right\|^2 , \tag{4.9}$$

which requires the computation of $S(u)^\top x$, which implies the solution of the adjoint problem. In our numerical experiments we have found that in general, the approximation (4.9) requires more realizations than the approximation (4.5) in order to achieve similar accuracy of the misfit (compare section 5).

Using the stochastic misfit approximation $\widehat{\mathsf{misfit}}(u)$ given by (4.5) or (4.9) we can now use the SAA method to solve the (stochastic) optimization problem and approximately recover the model. It is important to note that for this case, since the expected value is *inside* the norm, using SA methods in not justified.

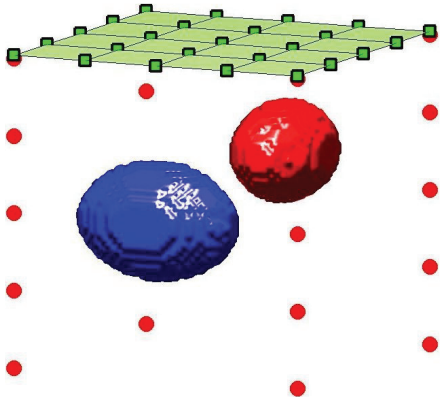

FIG. 2. *A sketch of the DC resistivity experiment. Sources in red are located in the boreholes, while the receivers (green) are on the surface. The red ellipsoid has a conductivity of* 10 *S/m and the blue has a conductivity of* 0.1 *S/m. Color is available only in the online version.*

5. Numerical experiments. In this section we perform numerical experiments to test our approximations and experiment with different methods and parameters. We use both the DC resistivity and the seismic tomography test problems. Our goals are to test the effectiveness of different methods for the solution of the same problem.

5.1. Experiments with the DC resistivity experiment. For the DC resistivity, the code for the forward problem and Jacobians is based on the software package [25] that can be obtained online.

Our goal here is to compare deterministic and stochastic methods. Since the deterministic method requires a large computer memory, we have chosen to use LBFGS for all methods such that the comparison is done on equal footing.

In our first set of experiments we use a mesh of 32^3 cells for the experiments and we assume we have sources and receivers located on the surface and in boreholes. A sketch of the experiment is presented in Figure 2. Our "true" solution consists of an ellipsoid conductor with conductivity of 10 S/m and a resistor with conductivity of 0.1 S/m. For simplicity we use smooth recovery and choose a regularizer of the form

$$R(u) = \frac{\rho}{2} u^\top \nabla_h^\top \nabla_h u,$$

where ∇_h is a discretization of the gradient operator and ρ is chosen such that we obtain a fit to the data based on the discrepancy principle [33].

5.1.1. Recovery when all sources share the same receivers. In the first experiment we assume we have 250 sources and 250 receivers and that all sources share the same receivers. This leads to the stochastic problem (2.3). To avoid the "inverse crime," we generate data for the true model using a fine grid of 64^3. We then use three methods to solve the inverse problem. In the first method we use the deterministic LBFGS method (for details see [24, 25]). This is a deterministic optimization problem which yields the exact solution of the stochastic programming problem. In a second experiment we chose five vectors w and use the SAA method to obtain a solution, and finally, we use the SA method, using a single realization at each step and replacing realizations as we progress. We plot the different models obtained by the experiment in Figure 3. As can be clearly observed, the results obtained by the stochastic methods are almost identical to the result of the deterministic optimization.

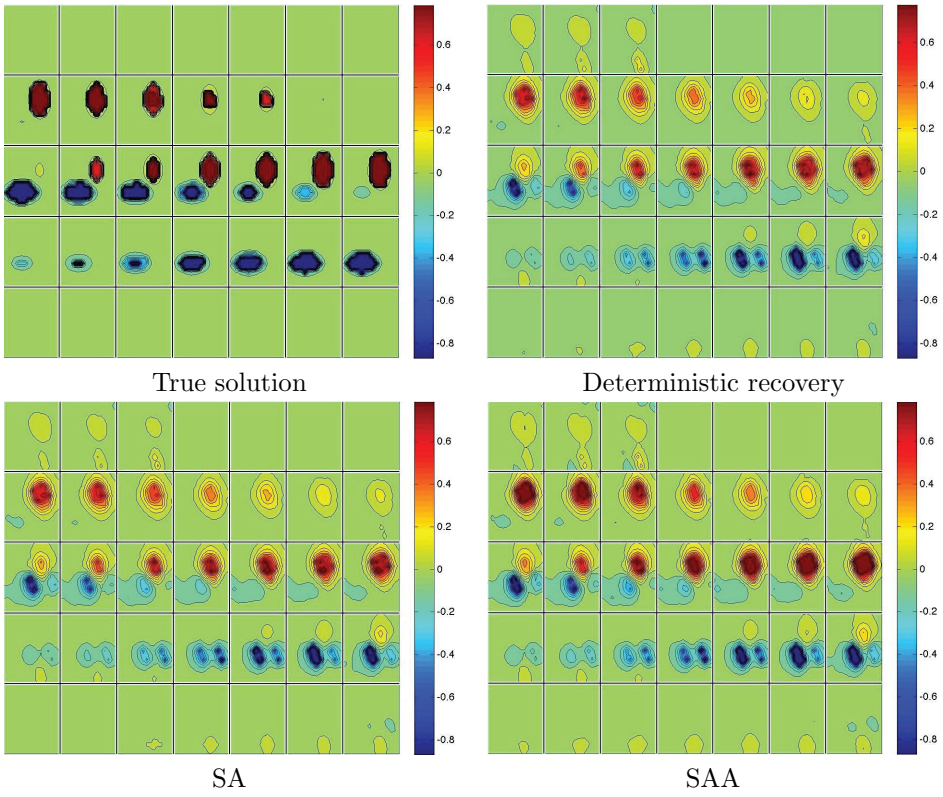

FIG. 3. *Comparison of the true model (top left), LBFGS recovery (top right), SA recovery (bottom left), and SAA recovery (bottom right). The three-dimensional models are sliced vertically and pieced together.*

In fact, the relative difference in the recovered models by the different techniques was well below 3%. In this example, we can compute the true value of the misfit and its stochastic approximation. The relative difference between the misfit obtained by the deterministic optimization method and the misfit obtained by the SAA method was 0.11%, and the difference between the misfit obtained by the SA method and the misfit obtain by the deterministic optimization was 0.13%. Thus, we were able to obtain excellent recovery of the model with very similar misfits using all three methods.

Here we add a few comments about the implementation of the different methods:

- For the deterministic and SAA optimization we stop when the norm of the gradient is 10^{-3} of its initial value.
- For the SA we use LBFGS and stop when the change in the solution is smaller than 10^{-3}.
- For the SAA we use only five vectors to obtain the solution. We then use the estimates developed in section 3 as a stopping criteria. When evaluating the gap (3.1) we solve the problem for five different samples starting from the solution obtained by the first sample. For all problems the solution for different samples converges in a single iteration, which indicates that we started very close to the solution and that the variability between solutions is small.
  The question of how many realizations of w's to use is answered by experimentation. For our starting point we experiment with the variability of the misfit.

TABLE 1
Number of iterations and PDE solves required by each of the methods. The difference between the solutions and the deterministic solution are recorded in the rightmost column.

Method	Iterations	PDE solves	Difference
Deterministic	51	51002	0%
SAA	43	860	2.1%
SA	325	650	1.2%

We have found that for a small set of w's (for example, using a single vector) we still get a reasonable approximation to the misfit (and a reasonable recovery of u), but repeating the experiment we found nonnegligible variance (of about 7%) between the solutions obtained from different realizations. When using five realizations we have found that the results did not change in a significant way (less than 3%) between different realizations. Determining the number of realizations a priori is an important question and will be studied elsewhere.

While the models recovered by each of the methods proposed are very similar in their quality, the computational complexity of obtaining these results varies widely. We estimate the computational cost by recording the number of PDE solves for each of the methods. The results are presented in Table 1. As table 1 shows, a factor of approximately 60 can be observed between the stochastic and deterministic methods. This is a remarkable factor that enables us to tackle larger problems in a fraction of the computational effort needed to solve the original problem.

5.1.2. Recovery when all sources do not share the same receivers. In the second set of experiments we compare the stochastic approximations given on (4.5) and (4.9) to solve the problem where the matrix $C \neq \mathbf{1}$. Three different cases are tested:

- The matrix C is dense and contains the standard deviation of each datum. We have assumed 1% noise and thus each entry in C is roughly 1% of the value of the corresponding datum.
- The matrix C is sparse, made of 0's and 1's, and contains only 10% nonzero entries. The entries are chosen at random.
- There are equal numbers of sources and receivers and the matrix C is diagonal, $C = I$, that is, each source has a single receiver. This case is common in some geophysical experiments when sources and receivers move simultaneously.

Three different optimization problems are solved for each case:

- We use a deterministic LBFGS to solve the problem.
- SAA with the stochastic approximation (4.5) is used (referred to as SAA1).
- SAA with the stochastic approximation (4.9) is used (referred to as SAA2).

For this experiment we use 500 sources and 500 receivers in the same configuration as above. Once again, we record the number of iterations and PDE solves for each of the cases. We also record the difference between the models recovered by stochastic programming to the models recovered by the deterministic one. The results for the different cases are summarized in Table 2. A few comments are in order.

- For SAA1 we used five vectors and found that it was sufficient to obtain results with little variability.
- For SAA2 we used two vectors for w_i and five vectors for x_j and the computation is almost double compared with SAA1. As can be seen in Table 2 SAA1 has a smaller variance compared with SAA2.

TABLE 2

Number of iterations and PDE solves required by each of the methods for different C's. The difference between the solutions and the deterministic solution are recorded in the rightmost column.

C	Method	Iterations	PDE solves	Difference
Dense	Deterministic	25	26500	0%
	SAA1	33	321	1.5%
	SAA2	38	654	5.1%
Sparse	Deterministic	31	33500	0%
	SAA1	32	325	1.7%
	SAA2	36	745	5.3%
Diagonal	Deterministic	27	28500	0%
	SAA1	32	331	0.9%
	SAA2	39	789	6.2%

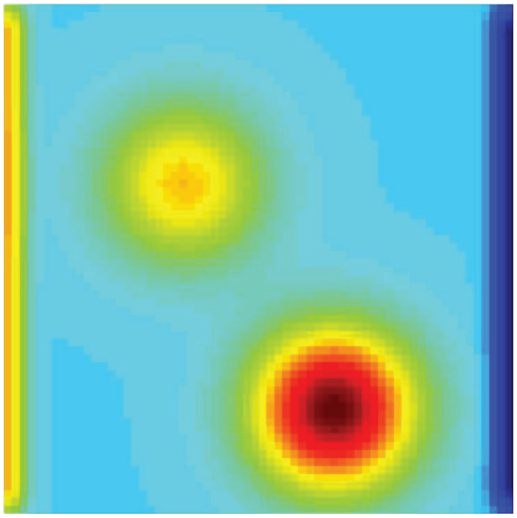

FIG. 4. *A sketch of the seismic tomography experiment. The parameter to be recovered is the seismic velocity, the sources (in the blue left region), and the receivers (in the yellow right region). Color is available only in the online version.*

- All optimization algorithms were terminated when the gradient was reduced to 10^{-3} of its original value.

As can be seen from Table 2, the model recovered by statistical methods is very similar to the models recovered by the deterministic models. This is done in a fraction of the cost of the deterministic method.

5.2. Experiments with seismic tomography. In our second experiment we use the two-dimensional seismic borehole tomography problem presented in Example 2. Similar problems in seismology are discussed in [8, 32]. In our experiment we use a single low frequency as a model problem. More complex problems (that require other sophistication such as continuation in frequency and mesh) are discussed in [8].

We assume two boreholes 1 meter apart and assume that the left borehole contains 500 sources while the right borehole contains 500 receivers. We use a frequency of 2 Hz, which for the material properties assumed here implies roughly three wavelengths between boreholes. A sketch of the experiment and the true model is plotted in Figure 4.

For this experiment we use two different meshes. We start with a mesh of 128^2 and then use a mesh of 256^2 to compare the scalability of our algorithm. In this

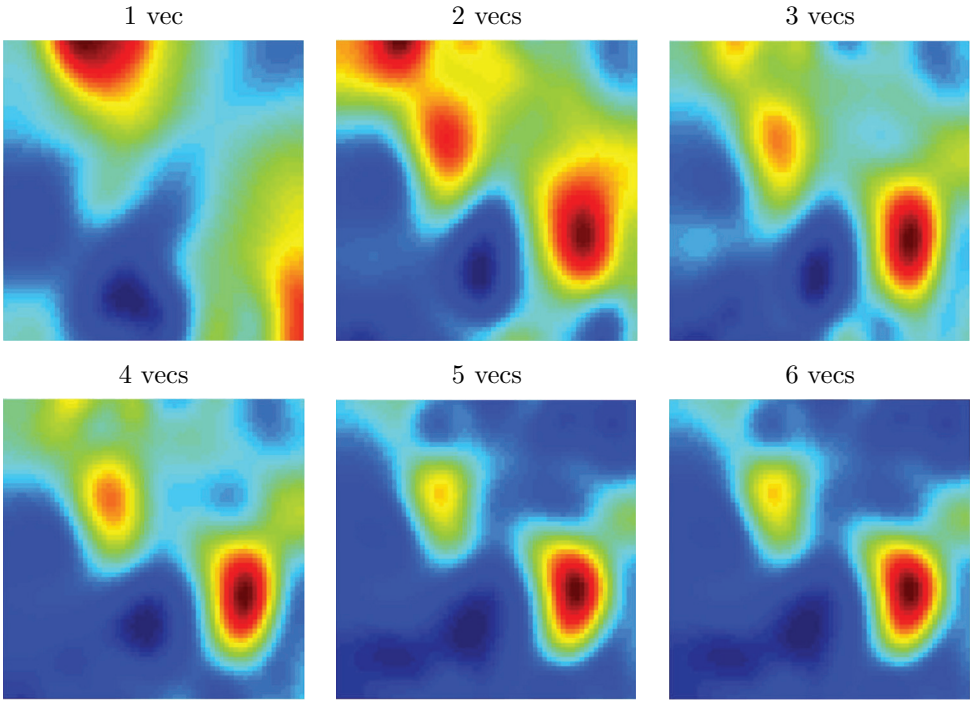

FIG. 5. *Models obtained for the seismic tomography experiment using one to six samples.*

TABLE 3

Number of iterations and PDE solves required by each of the methods. The difference between the solutions and the deterministic solution are recorded in the rightmost column.

Mesh # of realization	1	2	3	4	5	6
128^2	76	23	12	6	4	4
256^2	73	25	11	5	3	3

experiment, we use only SAA; however, unlike the previous example, where we set the number of vectors a priori, we use continuation to determine the number of samples needed. We start with a single sample, solve the optimization problem, and then use two new samples, starting from the previous solution. We continue until the recovered model does not change much (less that 1% change). In Figure 5 we plot the models obtained through this process. The number of LBFGS iterations on every level and mesh is recorded in Table 3.

As can be clearly seen, using a single sample (single simultaneous source) was not beneficial here. However, when using five and then six sources the result was unchanged. It is also evident that using "hot starts" helps in reducing the number of iterations needed for each new set of realizations. Finally, it can also be observed that the algorithm is unaffected by the mesh size. This implies that future work that combines Monte Carlo with multilevel techniques has merit. We believe that such algorithms that use continuation in the number of samples and mesh size may play a more important role in the future.

6. Discussion and summary. In this paper we explored the incorporation of stochastic programming techniques into PDE optimization problems with multiple

right-hand sides. We have shown the equivalence between the deterministic optimization problem and the stochastic ones. We then proposed stochastic programming techniques that allow for the solution of the stochastic problems in a fraction of the cost of deterministic techniques.

It is interesting to see that the stochastic programming problems are obtained by combining very basic linear algebra and probability. This allows us to understand and to extend the use of some of the methods previously proposed in the literature, the so-called simultaneous sources, that are based on physical arguments. In particular, we extended the method to deal with the case that not all sources share the same receivers, including the case where each source receiver combination has a different noise level.

There are a number of open questions that will be investigated in future work:

- How to choose the number of vectors in the SAA algorithm? The question can be formulated as, "How many vectors are needed in order to approximate the expected value?" We claim that this problem is similar in spirit to the question of how to discretize a continuous optimization problem into a discrete form. Two approaches have been taken in the literature. First, one can try to obtain a priori estimates of the error and discretize the optimization problem by using these estimates, and second, one can use a post priori estimate of the error and adaptively discretize the problem. The former is known to be pessimistic and the latter is known to be more efficient. We believe that a similar approach should be taken here and one should use adaptivity to estimate the number of vectors in SAA. We have experimented with this approach for the seismic tomography problem but would like to have a more quantitative evaluation.
- Given the fact that we know something about the distribution and structure of S, is there a better choice than the Rademacher distribution? The paper [29] suggests that in cases where we know something about the properties of the matrix S better approximation can be obtained.
- We intend to further study the approximation properties of the stochastic estimates (4.5) and (4.9).

There are a few obvious extensions to be done. First, for problems that contain only a small number of right-hand sides one can use algorithms that converge better and faster than LBFGS. In particular, the Gauss–Newton method and methods that are based on equality constrained optimization should have better numerical properties [13]. In the above experiments we avoided the use of these methods so we can compare the deterministic optimization to the two stochastic programming techniques. Using better optimization techniques can further deepen the gap between stochastic programming and deterministic optimization in the context of our problems. A second question is the performance of SA methods and the development of adequate theory for methods beyond steepest descent. Finally, applying the methods to different problems in science and engineering and speeding up practical inversion codes should be made.

REFERENCES

[1] V. Akcelik, G. Biros, and O. Ghattas, *Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation*, in Supercomputing, ACM/IEEE 2002 Conference, (2002), pp. 41–41.

[2] H. Avron, P. Maymounkov, and S. Toledo, *Blendenpik: Supercharging LAPACK's least-squares solver*, SIAM J. Sci. Comput., 32 (2010), pp. 1217–1236.

[3] Z. Bai, M. Fahey, and G. Golub, *Some large scale matrix computation problems*, J. Comput. Appl. Math., 74 (1996), pp. 71–89.

[4] C.J. Beasley, *A new look at marine simultaneous sources*, Leading Edge, 27 (2008), pp. 914–917.
[5] L.T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, eds., *Real-Time PDE-Constrained Optimization*, SIAM, Philadelphia, 2009.
[6] R. Bird, G. Chin, W. Nevleitt, and J. Nocedal, *On the use of stochastic Hessian information in unconstrained optimization*, submitted, 2010; available online from http://www.optimization-online.org/DB_FILE/2010/06/2657.pdf.
[7] M. Cheney, D. Isaacson, and J.C. Newell, *Electrical impedance tomography*, SIAM Rev., 41 (1999), pp. 85–101.
[8] I. Epanomeritakis, V. Akcelik, O. Ghattas, and J. Bielak, *A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion*, Inverse Problems, (2008).
[9] R. Fletcher, S. Leyffer, and P.L. Toint, *On the global convergence of a filter-SQP algorithm*, SIAM J. Optim., 13 (2002), pp. 44–59.
[10] G. Golub and U. Von Matt, *Generalized cross-validation for large scale problems*, J. Comput. Graph. Statist., 6 (1997), pp. 1–34.
[11] M.D. Gunzburger, *Prespectives in Flow Control and Optimization*, SIAM, Philadelphia, 2003.
[12] E. Haber and U. Ascher, *Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients*, SIAM J. Sci. Comput., 22 (2001), pp. 1943–1961.
[13] E. Haber, U. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.
[14] E. Haber, U. Ascher, and D. Oldenburg, *Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach*, Geophys., 69 (2004), pp. 1216–1228.
[15] F.J. Herrmann, Y.A. Erlangga, and T.T.Y. Lin, *Compressive simultaneous full-waveform simulation*, Geophys., 74 (2009), p. A35.
[16] M.F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Comm. Statist. Simulation Comput., 19 (1990), pp. 433–450.
[17] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, ADv. Des. Control 15, SIAM, Philadelphia, 2008.
[18] M.E. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.
[19] J.R. Krebs, J.E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M. Lacasse, *Fast full-wavefield seismic inversion using encoded sources*, Geophys., 74 (2009), pp. WCC177–WCC188.
[20] Y. Li and D.W. Oldenburg, *Inversion of 3-D DC resistivity data using an approximate inverse mapping*, Geophys. J. Internat., 116 (1994), pp. 557–569.
[21] J.J. Linderoth, A. Shapiro, and S. Wright, *The empirical behavior of sampling methods for stochastic programming*, Ann. Oper. Res., 142 (2006), pp. 215–241.
[22] S.A. Morton and C.C. Ober, *Faster shot-record depth migrations using phase encoding*, in Technical Program Expanded Abstracts 17, Society of Exploration Geophysicists, Tulsa, OK. 1998, pp. 1131–1134.
[23] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim., 19 (2008), pp. 1574–1609.
[24] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 1999.
[25] A. Pidlisecky, E. Haber, and R. Knight, *Resinvm3d: A MATLAB 3-D resistivity inversion package*, Geophys., 72 (2006), pp. 212–224.
[26] R.G. Pratt, *Seismic waveform inversion in the frequency domain, part* 1: *Theory, and verification in a physical scale model*, Geophys., 64 (1999), pp. 888–901.
[27] J. Romberg, R. Neelamani, C. Krohn, J. Krebs, M. Deffenbaugh, and J. Anderson, *Efficient seismic forward modeling and acquisition using simultaneous random sources and sparsity*, Geophys., to appear.
[28] L.A. Romero, D.C. Ghiglia, C.C. Ober, and S.A. Morton, *Phase encoding of shot records in prestack migration*, Geophys., 65 (2000), pp. 426–436.
[29] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, 1996.
[30] N.N. Schraudolph, J. Yu, and S. Günter, *A stochastic quasi-Newton method for online convex optimization*, J. Machine Learning Research, (2007), pp. 436–443.
[31] A. Shapiro, D. Dentcheva, and D. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, Philadelphia, 2009.
[32] T. van Leeuwen, A.Y. Aravkin, and F.J. Herrmann, *Seismic waveform inversion by stochastic optimization*, Int. J. Geophys, (2011), 689041.
[33] C. Vogel, *Computational Methods for Inverse Problem*, SIAM, Philadelphia, 2001.