STOCHASTIC ALGORITHMS FOR INVERSE PROBLEMS INVOLVING PDES AND MANY MEASUREMENTS

FARBOD ROOSTA-KHORASANI, KEES VAN DEN DOEL AND URI ASCHER *

Abstract.
Inverse problems involving systems of partial differential equations (PDEs) can be very expensive to solve numerically. This is so especially when many experiments, involving different combinations of sources and receivers, are employed in order to obtain reconstructions of acceptable quality. The mere evaluation of a misfit function (the distance between predicted and observed data) often requires hundreds and thousands of PDE solves. This article develops and assesses dimensionality reduction methods, both stochastic and deterministic, to reduce this computational burden.

We assume that all experiments share the same set of receivers and concentrate on methods for reducing the number of combinations of experiments, called simultaneous sources, that are used at each stabilized Gauss-Newton iteration. Algorithms for controlling the number of such combined sources are proposed and justified. Evaluating the misfit approximately, except for the final verification for terminating the process, always involves random sampling. Methods for selecting the combined simultaneous sources, involving either random sampling or truncated SVD, are proposed and compared. Highly efficient variants of the resulting algorithms are identified, and their efficacy is demonstrated in the context of the famous DC resistivity and EIT problems. We present in detail our methods for solving such inverse problems. These methods involve incorporation of a priori information such as piecewise smoothness, bounds on the sought conductivity surface, or even a piecewise constant solution.

Key words. inverse problem, stochastic algorithm, partial differential equation, many experiments, DC resistivity, EIT

AMS subject classifications. 65N21, 65C05

1. Introduction. Much recent attention has been given to the idea of applying stochastic approximations in order to speed up otherwise expensive computational processes. Here we consider the problem of recovering a model $\mathbf{m} \in \mathbb{R}^{l_m}$ from measurements $\mathbf{d}_i \in \mathbb{R}^l$, $i = 1, 2, \ldots, s$. For each i, the data is predicted as a function of $\mathbf{m}$ by a forward operator $\mathbf{F}_i$, and the goal is to find (or infer) $\mathbf{m} = \mathbf{m}^*$ such that the misfit function

$$\phi(\mathbf{m}) = \sum_{i=1}^{s} \|\mathbf{F}_i(\mathbf{m}) - \mathbf{d}_i\|^2 \tag{1.1}$$

is roughly at a level commensurate with the noise.[1] Now, introducing a random vector $\mathbf{w} = (w_1, \ldots, w_s)^T$, with the probability distribution of $\mathbf{w}$ chosen to satisfy

$$\mathbb{E}(\mathbf{w}\mathbf{w}^T) = I, \tag{1.2}$$

(with $\mathbb{E}$ denoting the expected value with respect to $\mathbf{w}$ and I the $s \times s$ identity matrix), we can write (1.1) as

$$\phi(\mathbf{m}) = \mathbb{E}\left(\| \sum_{i=1}^{s} w_i(\mathbf{F}_i(\mathbf{m}) - \mathbf{d}_i)\|^2 \right). \tag{1.3}$$

Next, in an iterative process for minimizing, or more precisely reducing (1.1) sufficiently, consider approximating the expectation value at iteration n by random sampling from a set of s_n vectors $\mathbf{w}$, with $s_n \leq s$ potentially satisfying $s_n \ll s$; see, e.g., [31, 24, 17].

In the class of problems we focus on, the sought model is a grid injection of a function $m(\mathbf{x})$ in two or three space dimensions. Furthermore, the forward operator involves an approximate solution

*Dept. of Computer Science, University of British Columbia, Vancouver, Canada. farbod/kvdoel/ascher@cs.ubc.ca This work was supported by NSERC Discovery Grant 84306.

[1] Throughout this article we use the ℓ_2 vector norm unless otherwise specified.

of a partial differential equation (PDE), or more generally, a system of PDEs. We write this in discretized form as

$$A(\mathbf{m})\mathbf{u}_i = \mathbf{q}_i, \quad i = 1, \ldots, s, \tag{1.4a}$$

where $\mathbf{u}_i \in \mathbb{R}^{l_u}$ is the ith field, $\mathbf{q}_i \in \mathbb{R}^{l_u}$ is the ith source, and A is a square matrix discretizing the PDE plus appropriate side conditions. Furthermore, there are given projection matrices P_i such that

$$\mathbf{F}_i(\mathbf{m}) = P_i\mathbf{u}_i = P_i A(\mathbf{m})^{-1}\mathbf{q}_i \tag{1.4b}$$

predicts the ith data set. Note that the notation (1.4a) reflects an assumption of linearity in $\mathbf{u}$ but not in $\mathbf{m}$.

There are several problems of practical interest in the form (1.1), (1.4), where the use of many experiments, resulting in a large number s (say, s is in the thousands), is crucial for obtaining credible reconstructions in practical situations. These include electromagnetic data inversion in mining exploration (e.g., [25, 13, 18, 27]), seismic data inversion in oil exploration (e.g., [15, 22, 30]), DC resistivity (e.g., [32, 29, 20, 19, 11]) and EIT (e.g., [6, 8]; see more specifically Example 5.5 in [12][2]). The last of these well-known problems enjoys the most extensive theory (e.g., [28, 1, 4]). This theory strongly suggests that many well-placed experiments are a practical must, especially when the surface $m(\mathbf{x})$ to be recovered is rough and has large gradients. Note that in this context, the mere evaluation of the objective function $\phi(\mathbf{m})$ of (1.1) involves solving s PDE systems. Thus, methods to alleviate the resulting computational burden are highly sought after.

A popular method of simultaneous random sources [31, 24, 30, 22, 19, 14] selects the w_i's in (1.3) to be randomly ± 1, which satisfies (1.2). This choice of $\mathbf{w}$ minimizes variance of the residual approximating ϕ [23]. A different set of $\hat{s}$ realizations ($\hat{s} \ll s$) is chosen at each iteration, combined with an averaging procedure for the measurements. However, more recently doubts were cast as to the wisdom of choosing $w_i = \pm 1$. In [5], the authors consider lower bounds on $\hat{s}$ in order to achieve a fixed probabilistic error. Based on this, drawing $\mathbf{w}$ from the standard normal distribution is no longer a clearly inferior method. Furthermore, a deterministic simultaneous sources method to reduce dimensionality may be obtained upon applying truncated singular value decomposition (TSVD) to the data re-cast as an $l \times s$ matrix. Our first contribution in the present paper is to discuss and compare these three simultaneous sources methods for choosing the weights $\mathbf{w}$.

The papers cited above all appear to assume one purpose for the approximate evaluation of the misfit function $\phi(\mathbf{m})$. In contrast, here we identify three different reasons or purposes for this task, and furthermore we show that these different purposes may well require different estimation methods.

Applying these simultaneous sources methods for the problem described in (1.4) yields an efficient algorithm if the data in different experiments are measured at the same locations, i.e., $P_i = P \ \forall i$. This is because in such a case we have

$$\sum_{i=1}^{s} w_i\mathbf{F}_i = \sum_{i=1}^{s} w_i P_i A(m)^{-1}\mathbf{q}_i = PA(m)^{-1}\big(\sum_{i=1}^{s} w_i\mathbf{q}_i\big), \tag{1.5}$$

which can be computed with a single PDE solve per realization of a vector of weights $\mathbf{w}$. In [11] the authors proposed a method that does not suffer from such a restriction on the receiver locations, whereby (1.3) is estimated by using for $\mathbf{w}$ vectors with a single nonzero component at a random location which is scaled to satisfy (1.2) (cf. [5]). This choice allows the computation of $\sum_{i=1}^{s} w_i\mathbf{F}_i$ in a single PDE solve per realization; it boils down to selecting a random subset of the given experiments at each iteration, rather than their weighted combination.

[2] See also the Wikipedia description for electrical impedance tomography.

In this paper we assume that $P_i = P,\ i = 1, \ldots, s$. For this case we therefore have in the above descriptions four methods for choosing the weights, which may be fused and compared. They are described in detail in Section 3. It is important, especially when using stochastic weighting methods, to carefully derive merit criteria for the minimization iterations, for one thing because evaluating (1.1) too often defeats the purpose of reducing the number of PDE evaluations. This is addressed in Section 3 as well.

The question of selecting the sample size s_n is addressed in Section 4. We propose two new algorithms which allow s_n to be very small for small n, and potentially significantly increase as the iteration approaches a solution. Algorithm 1 in Section 4.1 has the advantage of being simple, and it generates an exponentially increasing sequence of s_n values. Algorithm 2 in Section 4.2 uses cross validation in a manner similar to but not the same as that proposed in [11], and it generates a potentially more moderately increasing sequence of s_n values. The latter algorithm is particularly useful when s is "too large" in the sense that even near a satisfactory solution for the given inverse problem, far fewer than s experiments are required to satisfy the given error tolerances, a situation we qualitatively refer to as *embarrassing redundancy*. Within the context of these two algorithms, we compare the resulting weighting methods of Section 3.1 against the more generally applicable random subset method proposed in [11], and find that the three simultaneous sources methods are roughly comparable and are better than the random subset method by a factor of roughly 2 on average.

The computational work in Section 5 is done in the context of an EIT, or DC resistivity, problem. This is a simpler forward problem than low-frequency Maxwell's equations, and yet it reflects a similar spirit and general behaviour, allowing us to concentrate on the issues in focus here. A description of the implementation details is given in Appendix A. Conclusions and plans for future work are offered in Section 6.

2. Model reduction for the inverse problem. Let us suppose for now that the forward operators $\mathbf{F}_i(\mathbf{m})$, each involving a PDE solution, are given as in (1.4): see Appendix A and Section 5 for a specific instance, used for our numerical experiments. Next, consider the problem of minimizing the misfit function $\phi(\mathbf{m})$ defined in (1.1). With the sensitivity matrices

$$J_i(\mathbf{m}) = \frac{\partial \mathbf{F}_i}{\partial \mathbf{m}}, \quad i = 1, \ldots, s, \tag{2.1}$$

we have the gradient

$$\operatorname{grad} \phi(\mathbf{m}) = 2 \sum_{i=1}^{s} J_i^T (\mathbf{F}_i(\mathbf{m}) - \mathbf{d}_i). \tag{2.2}$$

An iterative method such as Gauss-Newton (GN) is typically designed to reduce the misfit function using the gradient. Although the methods and issues under consideration here do not require a specific optimization method we employ variants of the GN method throughout the present article, thus achieving a context in which to focus our attention on the new aspects of this work and enabling comparison to past efforts. In particular, the way in which the GN method is modified next is important more generally.

Our purpose is to use on average far fewer than $2s$ evaluations of forward operator components $\mathbf{F}_i(\mathbf{m})$ per iteration. In order to develop this, let us rewrite the objective function (1.1) using the Frobenius norm as

$$\begin{aligned} \phi(\mathbf{m}) &= \|F(\mathbf{m}) - D\|_F^2, \\ & F = \big[\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_s\big] \in \mathbb{R}^{l \times s}, \quad D = \big[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_s\big] \in \mathbb{R}^{l \times s}. \end{aligned} \tag{2.3a}$$

Note that upon using our assumption that $P_i = P\ \forall i$, we can write (1.4b) as

$$F = PA^{-1}(m)Q, \quad Q = \big[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_s\big] \in \mathbb{R}^{l_u \times s}. \tag{2.3b}$$

Next, in the nth iteration, $1 \le n \le N$, consider approximating ϕ of (2.3), replacing the s experiments by s_n experiments ($s_n \le s$) with linear combinations of the sources, described by the $s \times s_n$ matrix $W = W_n = \left[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{s_n}\right]$. We define the obtained estimator of (2.3) as

$$\hat{\phi}(\mathbf{m}, W) = \frac{1}{s_n} \|(F(\mathbf{m}) - D)W\|_F^2. \tag{2.4}$$

Choices of W which keeps $\hat{\phi}(\mathbf{m}, W)$ close to $\phi(\mathbf{m})$ are discussed in Section 3.1. Now it is also easy to return to a form like (1.1) for $\hat{\phi}(\mathbf{m}, W)$ and define sensitivity matrices $\hat{J}_i = J_i(\mathbf{m}, W)$ and gradient $\text{grad}_{\mathbf{m}}\hat{\phi} = \text{grad}_{\mathbf{m}}\,\hat{\phi}(\mathbf{m}, W)$ analogously to (2.1) and (2.2), respectively. The GN iteration for (2.4) at a current iterate $\mathbf{m} = \mathbf{m}_n$ with $W = W_n$ calculates the correction as the solution of the linear system

$$\left(\sum_{i=1}^{s_n} \hat{J}_i^T \hat{J}_i\right) \delta\mathbf{m} = -\,\text{grad}_{\mathbf{m}}\,\hat{\phi}, \tag{2.5a}$$

followed by the update

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \delta\mathbf{m}. \tag{2.5b}$$

Several nontrivial modifications are required to adapt this prototype method for our purposes, and these are described in context in Appendix A, resulting in a method we refer to as *stabilized GN*. This method replaces the solution of (2.5a) by r preconditioned conjugate gradient (PCG) inner iterations, which costs $2r$ solutions of the forward problem per iteration, for a moderate integer value r. Thus, if N outer iterations are required to obtain an acceptable solution then the total work estimate (in terms of the number of PDE solves) is approximated *from below* by

$$\text{Work estimate} = 2(r+1)\sum_{n=1}^{N} s_n. \tag{2.6}$$

This indicates how keeping s_n small is important; see [11].

Note that an alternative method to GN such as L-BFGS [26] would require only $r = 1$ in (2.6). However, the number of such iterations would be significantly higher. This point again does not affect the issues addressed in this article and is not pursued further.

3. Choosing weights and stopping criteria. In Section 3.1 below we discuss our four methods for choosing matrices W in (2.4). Then, in Section 3.2 we address two purposes for approximating $\phi(\mathbf{m})$ which are additional to that used in (2.5).

3.1. Selecting a sampling method. In this section we discuss two approaches for choosing $W = W_n$ in the nth stabilized GN iteration (2.5): a stochastic approach where each of the s_n columns of W represents a realization of the weight vector $\mathbf{w}$ which appears in (1.3), drawn from an appropriate probability distribution; and a deterministic approach.

Consider first the stochastic approach. In the sequel, we consider each column $\mathbf{w} = \mathbf{w}_j$ of W to be independently drawn from (a) the Rademacher distribution, where $Pr(w_i = 1) = Pr(w_i = -1) = \frac{1}{2}$ (referred to in what follows as *Hutchinson*, in deference to [23, 5]), (b) the standard normal distribution (referred to as *Gaussian*), and (c) uniformly from the column vectors of the scaled identity matrix (referred to as *Random Subset*). This third method [11] can be efficiently applied to estimating (1.3) even in the case where receivers are not shared.

The second approach is to abandon stochastic optimization altogether in this section, and instead select W deterministically. In order to do that, we use the well-known TSVD approach. Thus, let $D = U\Sigma V^T$, where Σ is the diagonal matrix of singular values in decreasing order, and U and V are

orthogonal matrices. Note that $DV = U\Sigma$. Subsequently, we approximate (2.3) by letting $W = W_n$ consist of the first s_n columns of V. This then uses only the first s_n columns of $U\Sigma$ and QV, and should bring out the essence of what is in the data, especially when the current iterate is far from the solution of the inverse problem. A plot of the singular values for a typical experiment (in the present context) is depicted in Figure 3.1. The quick drop in the singular values suggests that just

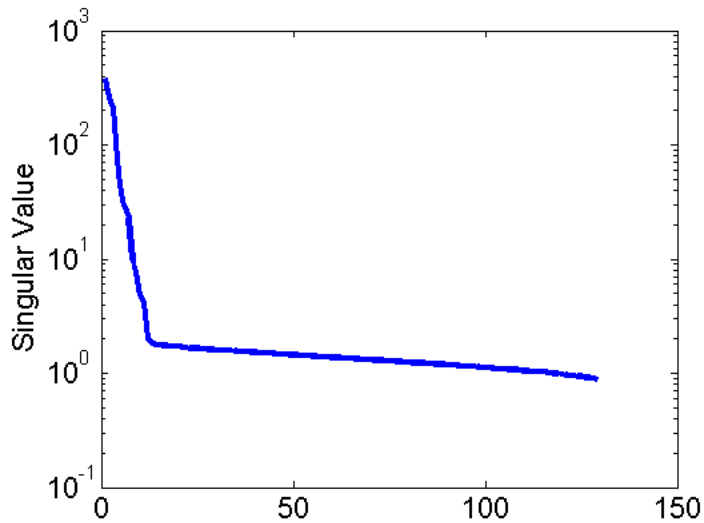


FIG. 3.1. *The singular values of the data used in Example 2 of Section 5.*

a few singular vectors (the first columns of the orthogonal matrix U) represent the entire data well. This simple method is suitable when both dimensions of the data matrix D are not too large. The SVD is performed only once. Then, in the nth stabilized GN iteration, the first s_n columns of V provide fixed and deterministic weights for this simultaneous sources method.

3.2. Stopping criterion and uncertainty check. In this paper we make the popular assumption that there is a known tolerance ρ such that the stopping criterion is given by

$$\phi(\mathbf{m}) \leq \rho, \tag{3.1a}$$

with $\phi(\mathbf{m})$ not being much smaller than ρ. For instance, consider the simplest (and often unrealistic) case where for all experiments there is a Gaussian noise distribution for which the (same) standard deviation σ is known. Thus $D = D^* + \sigma\mathcal{N}$, where $D^* = F(\mathbf{m}^*)$, with $\mathcal{N}$ an $l \times s$ matrix of i.i.d Gaussians. Using the discrepancy principle (see, e.g., [33]) we wish to terminate the algorithm when (2.3) falls below some multiple $\eta \gtrapprox 1$ of the noise level squared, i.e. $\sigma^2\|\mathcal{N}\|_F^2$. Since the noise is not known, we replace $\|\mathcal{N}\|_F^2$ by its expected value, sl, obtaining

$$\rho = \eta\sigma^2 sl. \tag{3.1b}$$

Unfortunately, however, the mere calculation of $\phi(\mathbf{m})$ requires s PDE solves. We therefore wish to perform this check as rarely as possible. Fortunately, we often have in $\hat{\phi}(\mathbf{m}, W)$ a good, unbiased estimator of $\phi(\mathbf{m})$ using stochastic weight matrices $W = W^e$ with far fewer columns than s, provided the columns of W^e are independent and satisfy (1.2). Thus, in the course of an iteration we can perform the relatively inexpensive *uncertainty check* whether

$$\hat{\phi}(\mathbf{m}, W^e) \leq \rho. \tag{3.2}$$

This is like the stopping criterion, but in expectation (with respect to W^e). If (3.2) is satisfied, it is an indication that (3.1) is likely to be satisfied as well, so we check the expensive (3.1a) only then.

Note that, since we want $\hat{\phi}(\mathbf{m}, W^e)$ to be an unbiased estimator, W^e should not be constructed deterministically! Furthermore, it should be independent of the noise. In our experiments we use the Hutchinson method, because its trace estimator has the smallest variance [23].

4. Adaptive selection of sample size. In this section we describe two algorithms for determining the sample size s_n in the nth stabilized GN iteration. Algorithm 1 adapts s_n in a brute force manner. Algorithm 2 uses a cross validation technique to avoid situations in which s_n grows too rapidly or becomes larger than necessary.

4.1. Sample size selection using uncertainty checks. While the management strategy of s_n in this algorithm is simply to increase it so long as (3.1) is not met, its novelty lies in the fusion of different strategies for selecting the weight matrices at different stages of each iteration. Our algorithm consists of three main steps: (i) *data fitting* – a stabilized GN outer iteration (2.5); (ii) *uncertainty check* – a check for condition (3.2); and (iii) depending on the outcome of the uncertainty check, perform either *sample size adjustment* or *stopping criterion* check for termination.

Algorithm 1 Solve inverse problem using uncertainty check

Given: sources $Q = [\mathbf{q}_1\mathbf{q}_2 \cdots \mathbf{q}_s]$, measurements $D = [\mathbf{d}_1\mathbf{d}_2 \cdots \mathbf{d}_s]$, stopping criterion level ρ (i.e. the desired misfit) and initial guess $\mathbf{m}_0$.
Initialize: $\mathbf{m} = \mathbf{m}_0$, $s_0 = 1$.
for $n = 0, 1, 2, \cdots$ until termination **do**
- Choose a $W_n^f \in \mathbb{R}^{s \times s_n}$ deterministically or stochastically as described in Section 3.1.
- **Fitting**: Perform one stabilized GN iteration approximating (2.5), with $W = W_n^f$.
- Choose $W_n^e \in \mathbb{R}^{s \times s_n}$ stochastically as described in Section 3.2.
- **Uncertainty Check**: Compute (3.2) using $\mathbf{m}_{n+1}$ and W_n^e.

if Uncertainty Check holds **then**
- **Stopping Criterion**: Compute (3.1) with $\mathbf{m}_{n+1}$. **Terminate** if it holds.

else
- **Sample Size Increase**: Increase s_{n+1}, for example set $\mathbf{s}_{n+1} = \min(2\mathbf{s}_n, s)$.

end if
end for

The exponential growth of the sample size in Algorithm 1 can be theoretically appealing, as such a schedule (unlike keeping s_n fixed) enables the general convergence theory of [16]. However, in cases where there is embarrassing redundancy in the set of experiments, it may not be desirable for the sample size to grow so rapidly and in an unchecked manner, as we could end up using far more experiments than what is actually needed. Some mechanism is required to control the growth of sample size, and one such is proposed next.

4.2. Adaptive selection of sample size using cross validation. For monitoring the growth of s_n more closely, one strategy is to compare the objective function ϕ at the current iterate to its value in the previous iterate, increasing the sample size if there is no sufficient decrease. Unfortunately, evaluating ϕ involves solving s PDEs, which defeats the purpose (in Section 5 typically the total numbers of PDE solves are small multiples of just one evaluation of ϕ). Fortunately, however, using an unbiased estimator of the objective function, described in Section 3.2, we can get a handle of how the objective function is likely to behave. A method of this sort, based on "cross validation", is proposed in [11] together with a Random Subset method for W. Here we generalize and adapt this technique in the present context. Thus, the following algorithm involves the steps of Algorithm 1, with an additional check for a sufficient decrease in the estimate (2.4) using another, independently selected weight matrix. Only in case that this test is violated, we increase the sample size.

Algorithm 2 Solve inverse problem using uncertainty check and cross validation

Given: sources $Q = [\mathbf{q}_1\mathbf{q}_2\cdots\mathbf{q}_s]$, measurements $D = [\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_s]$, stopping criterion level ρ (i.e. the desired misfit), objective function sufficient decrease factor $\kappa < 1$, and initial guess $\mathbf{m}_0$
Initialize: $\mathbf{m} = \mathbf{m}_0$, $s_0 = 1$.
for $n = 0, 1, 2, \cdots$ until termination **do**
- Choose a $W_n^f \in \mathbb{R}^{s\times s_n}$ deterministically or stochastically as described in Section 3.1.
- **Fitting**: Perform one stabilized GN iteration approximating (2.5), with $W = W_n^f$.
- Choose a $W_n^c \in \mathbb{R}^{s\times s_n}$ and $W_n^e \in \mathbb{R}^{s\times s_n}$ stochastically as described in Section 3.2.

if $\hat{\phi}(\mathbf{m}_{n+1}, W_n^c) \le \kappa\hat{\phi}(\mathbf{m}_n, W_n^c)$, i.e., **Cross Validation** is satisfied **then**
- **Uncertainty Check**: Compute (3.2) using $\mathbf{m}_{n+1}$ and W_n^e.

if Uncertainty Check holds **then**
- **Stopping Criterion**: Compute (3.1) with $\mathbf{m}_{n+1}$. **Terminate** if it holds; otherwise set $s_{n+1} = s_n$.

end if
else
- **Sample Size Increase**: for example, set $\mathbf{s}_{n+1} = \min(2\mathbf{s}_n, s)$.

end if
end for

Note that our use of the term "cross validation" does not necessarily coincide with its usual meaning in statistics. But the procedure retains the sense of a control set and this name is convenient. The performance of Algorithm 2 is not automatically better than that of Algorithm 1, as the computations in Section 5 demonstrate. However, it provides an important safety mechanism.

5. Numerical experiments. Our experiments are performed in the context of solving the EIT/DC resistivity problem. Consider a linear PDE of the form

$$\nabla\cdot\,(\sigma(\mathbf{x})\,\mathrm{grad}\,u) = q(\mathbf{x}), \quad \mathbf{x}\in\Omega, \tag{5.1a}$$

where $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, and σ is a conductivity function which may be rough (e.g., discontinuous) but is bounded away from 0: there is a constant $\sigma_0 > 0$ such that $\sigma(\mathbf{x}) \ge \sigma_0$, $\forall\mathbf{x}\in\Omega$. This elliptic PDE is subject to the homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial n} = 0, \quad \mathbf{x}\in\partial\Omega. \tag{5.1b}$$

For Ω we will consider a unit square or a unit cube. The inverse problem is to recover σ in Ω from sets of measurements of u on the domain's boundary for different sources q. This is a notoriously difficult problem in practice, so in Appendix A we detail the numerical methods employed both for defining the predicted data F and for solving the inverse problem in appropriately transformed variables.

5.1. Numerical experiments setup. The experimental setting we use is as follows: for each experiment i there is a positive unit point source at $\mathbf{x}_1^i$ and a negative sink at $\mathbf{x}_2^i$, where $\mathbf{x}_1^i$ and $\mathbf{x}_2^i$ denote two locations on the boundary $\partial\Omega$. Hence in (5.1a) we must consider sources of the form $\mathbf{q}_i(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{x}_1^i) - \delta(\mathbf{x}-\mathbf{x}_2^i)$, i.e., a difference of two δ-functions.

For our experiments in 2D, when we place a source on the left boundary, we place the corresponding sink on the right boundary in every possible combination. Hence, having p locations on the left boundary for the source would result in $s = p^2$ experiments. The receivers are located at the top and bottom boundaries. No source or receiver is placed at the corners.

In 3D we use an arrangement whereby four boreholes are located at the four edges of the cube, and source and sink pairs are put at opposing boreholes in every combination, except that there are

no sources on the point of intersection of boreholes and the surface, i.e., at the top four corners, since these four nodes are part of the surface where data values are gathered.

In the sequel we generate data $\mathbf{d}_i$ by using a chosen true model (or ground truth) and a source-receiver configuration as described above. Since the field u from (5.1) is only determined up to a constant, only voltage differences are meaningful. Hence we subtract for each i the average of the boundary potential values from all field values at the locations where data is measured. As a result each row of the projection matrix P has zero sum. This is followed by peppering these values with additive Gaussian noise to create the data $\mathbf{d}_i$ used in our experiments. Specifically, for an additive noise of 3%, say, denoting the "clean data" $l \times s$ matrix by D^*, we reshape this matrix into a vector $\mathbf{d}^*$ of length sl, calculate the standard deviation $\mathtt{sd} = .03\|\mathbf{d}^*\|/\sqrt{sl}$, and define $D = D^* + \mathtt{sd} * \mathtt{randn}(\mathtt{l}, \mathtt{s})$ using MATLAB's random generator function `randn`.

For all numerical experiments, the "true field" is calculated on a grid that is twice as fine as the one used to reconstruct the model. For the 2D examples, the reconstruction is done on a uniform grid of size 64^2 with $s = 961$ experiments in the setup described above, and we used $\eta = 1.2$. For our 3D examples, the reconstruction is done on a uniform grid of size 17^3 with $s = 512$ experiments, and we set $\eta = 1.5$.

In Section 5.2 below, for the first three examples we use the transfer function (6.3) with $\sigma_{\max} = 1.2 \max \sigma(\mathbf{x})$, and $\sigma_{\min} = .83 \min \sigma(\mathbf{x})$. In the ensuing calculations we then "forget" what the exact $\sigma(\mathbf{x})$ is. Further, we set the PCG iteration limit to $r = 20$, and the PCG tolerance to 10^{-3}. The initial guess is $\mathbf{m}_0 = \mathbf{0}$. Our last example is carried out using the level set method (6.4). Here we can set $r = 5$, significantly lower than above. The initial guess for the level set examples is displayed in Figure 5.1.

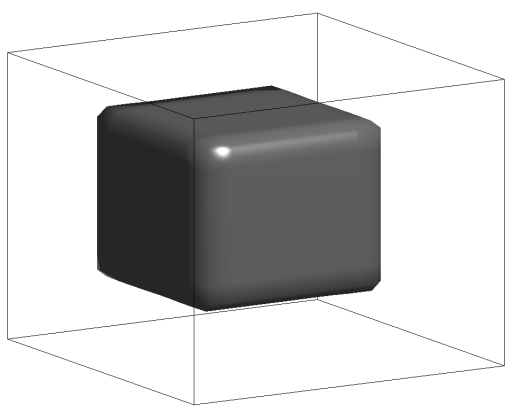


FIG. 5.1. *Example 4 – initial guess for the level set method.*

In addition to displaying the log conductivities (i.e., $\log(\sigma)$) for each reconstruction, we also show the log-log plot of misfit on the entire data (i.e. $\|F(\mathbf{m}) - D\|_F$) vs. PDE count. A table of total PDE counts (not including what extra is required for the plots) for each method is displayed. In this table, as a point of reference, we also include the total PDE count using the "plain vanilla" stabilized Gauss-Newton method which employs the entire set of experiments at every iteration.

We emphasize that, much as the rows in the work-unit table are easier to examine in order to determine which method is more efficient, it is important to also consult the corresponding data misfit plots, especially when the comparison is between relatively close quantities. This is so because one evaluation of the stopping criterion consumes a significant fraction of the total PDE count in each case, so an extra check that can randomly occur for a given experiment in one method and not

another may affect the work table far more than the misfit figures. In particular, the performance of the Hutchinson vs. Gauss estimators was found to be comparable in almost all experiments below.

Finally, before we turn to the numerical results let us comment on the expected general quality of such reconstructions. The quantifiers "good" and "acceptable" are relative concepts here. Our 3D experiments mimic DC geophysics surveys, where a reconstruction is considered good and acceptable if it generally looks like the true model, even remotely so. This is very different from the meaning of similar quantifiers in image denoising, for instance.

5.2. Numerical experiments comparing eight method variants. In each of the four examples below we apply Algorithm 1 and Algorithm 2 with $\kappa = 1$; smaller values of κ would result in more aggressive increases of the sample size between one stabilized GN iteration and the next.

Furthermore, for convenience of cross reference, we gather all resulting eight work counts in Table 5.1 below. The corresponding entries of this table should be read together with the misfit plots for each example, though.

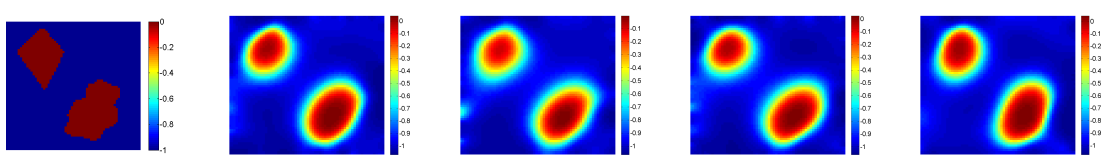
Example	**Algorithm**	Vanilla	Random Subset	Hutchinson	Gaussian	TSVD
1	1	86,490	3,788	1,561	1,431	2,239
	2		3,190	2,279	1,618	2,295
2	1	128,774	5,961	3,293	3,535	3,507
	2		3,921	2,762	2,247	2,985
3	1	36,864	6,266	1,166	1,176	1,882
	2		11,983	3,049	2,121	2,991
4	1	45,056	1,498	1,370	978	1,560
	2		2,264	1,239	896	1,656

TABLE 5.1

Work in terms of number of PDE solves for Examples 1–4. The "Vanilla" count is independent of the algorithms described in Section 4.

EXAMPLE 1.

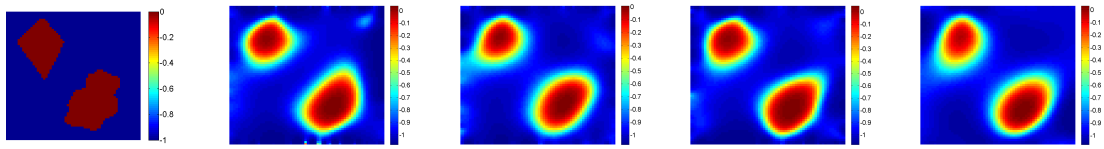
In this example, we place two target objects of conductivity $\sigma_I = 1$ in a background of conductivity $\sigma_{II} = 0.1$, and 3% noise is added to the data: see Figure 5.2(a). The reconstructions in Figures 5.2 and 5.3 are comparable.



(a) True model (b) Random Subset (c) Gaussian (d) Hutchinson (e) TSVD

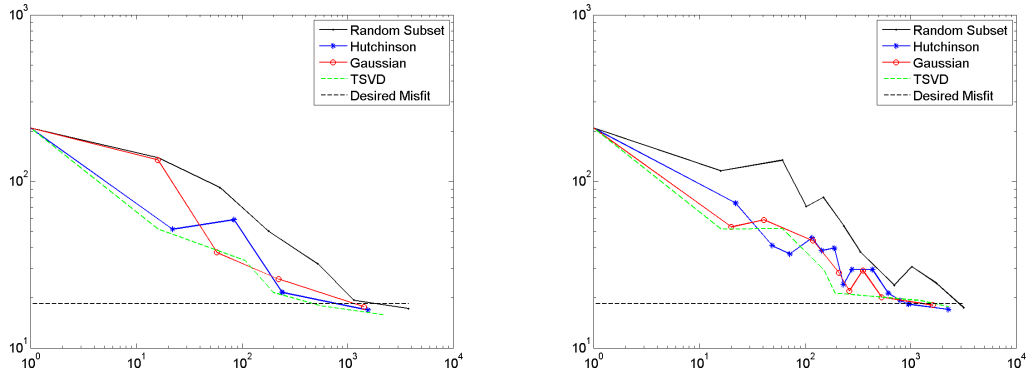
FIG. 5.2. *Example 1 – reconstructed log conductivity using Algorithm 1 and the four methods of Section 3.1.*

From Table 5.1 we see that all our methods offer vast improvements over the plain Vanilla method. Furthermore, the Random Subset method reduces the objective (i.e., misfit) function at a slower rate, requiring roughly twice as many PDE solves compared to the other methods of Section 3.1. Consulting also Figure 5.4, observe in addition that although the final PDE count for TSVD is slightly larger than for Hutchinson and Gaussian, it reduces the misfit at a faster, though comparable, rate. In fact, if we were to stop the iterations at higher noise tolerances then the TSVD method would have outperformed all others. In repeated similar tests, we have observed that the performance of Hutchinson and Gaussian is comparable.



(a) True model (b) Random Subset (c) Gaussian (d) Hutchinson (e) TSVD

FIG. 5.3. *Example 1 – reconstructed log conductivity using Algorithm 2 and the four methods of Section 3.1.*



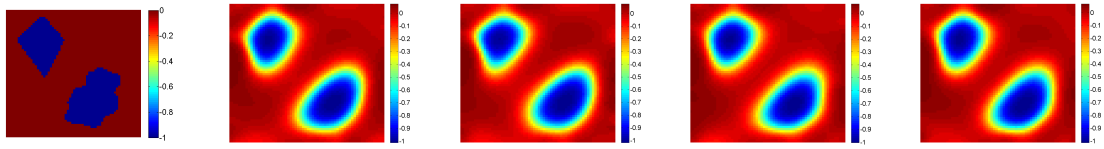
(a) Algorithm 1 (b) Algorithm 2

FIG. 5.4. *Data misfit vs. PDE count for Example 1.*

Finally, comparing the first two rows of Table 5.1 and the subplots of Figure 5.4, it is clear that the performance of Algorithms 1 and 2 is almost the same.

EXAMPLE 2.

For this example, we merely swap the conductivities of the previous one, see Figure 5.5(a), and add the lower amount of 1% *noise to the "exact data". The reconstruction results in Figures 5.5 and 5.6 are comparable. The performance indicators are gathered in Table 5.1 and Figure 5.7.*



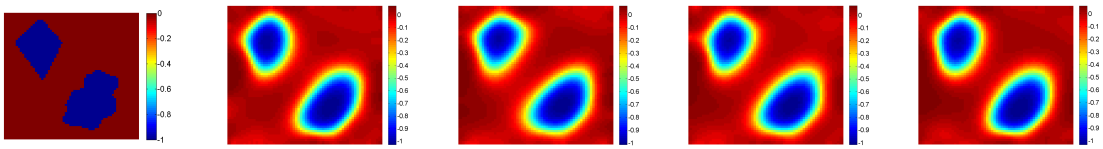
(a) True model (b) Random Subset (c) Gaussian (d) Hutchinson (e) TSVD

FIG. 5.5. *Example 2 – reconstructed log conductivity using Algorithm 1 and the four methods of Section 3.1.*

Note that since in this example the noise is reduced compared to the previous one, more PDE solves are required. Similar observations to all those made for Example 1 apply here as well, except that using the cross validation algorithm results in a notable reduction in PDE solves.

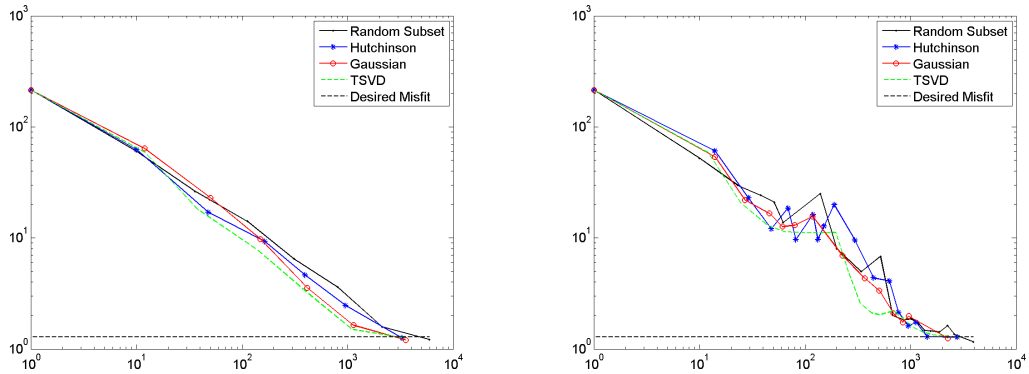
EXAMPLE 3.

In this 3D example, we place a target object of conductivity $\sigma_I = 1$ *in a background with conductivity* $\sigma_{II} = 0.1$. *See Figure 5.8, whose caption also explains what other plots for 3D runs depict. A* 2% *noise is added to the "exact" data.*



(a) True model (b) Random Subset (c) Gaussian (d) Hutchinson (e) TSVD

FIG. 5.6. *Example 2 – reconstructed log conductivity using Algorithm 2 and the four methods of Section 3.1.*



(a) Algorithm 1 (b) Algorithm 2

FIG. 5.7. *Data misfit vs. PDE count for Example 2.*

The reconstruction quality for all eight method variants, see Figures 5.9 and 5.10, appears less clean than in our other examples; however, the methods are comparable in this regard, which allows us to concentrate on their comparative efficiency. It should be noted that no attempt was made here to "beautify" these results by post-processing, a practice not unheard of for hard geophysical inverse problems. Better reconstructions are obtained in the next example which employs more a priori information and higher contrast.

In cases where more experiments are needed, the differences among the sampling methods are even more pronounced. This 3D example is one such case. All of the methods (excluding Vanilla) ended up using half of the experiments (i.e., $S_N \approx .5s$*) before termination. Clearly, the Random Subset method is far outperformed by the other three, see Table 5.1 and Figure 5.13.*

This is one example where Algorithm 1 achieves reconstructions of similar quality but more cheaply than Algorithm 2. This is so because in this case there is little embarrassing redundancy, i.e., larger sample sizes are needed to achieve the desired misfit, hence growing the sample size at a faster rate leads to an efficient algorithm. The sample size using cross validation grows more slowly, and relatively many GN iterations are performed using small sample sizes where each iteration decreases the misfit only slightly. These added iterations result in larger total PDE solve count.

EXAMPLE 4. *This one is the same as Example 3, except that we assume that additional prior information is given, namely, that the sought model consists of piecewise constant regions with conductivity values* σ_I *and* σ_{II}*. This mimics a common situation in practice. So we reconstruct using the level set method (6.4), which significantly improves the quality of the reconstructions: compare Figures 5.11 and 5.12 to Figures 5.9 and 5.10.*

Here we observe less difference among the various methods. Specifically, in repeated experiments, the Random Subset method is no longer clearly the worst, see Table 5.1 and Figure 5.14. The numbers

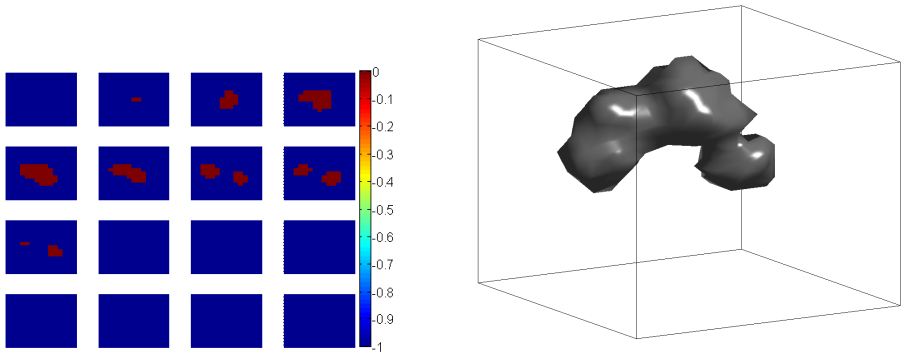
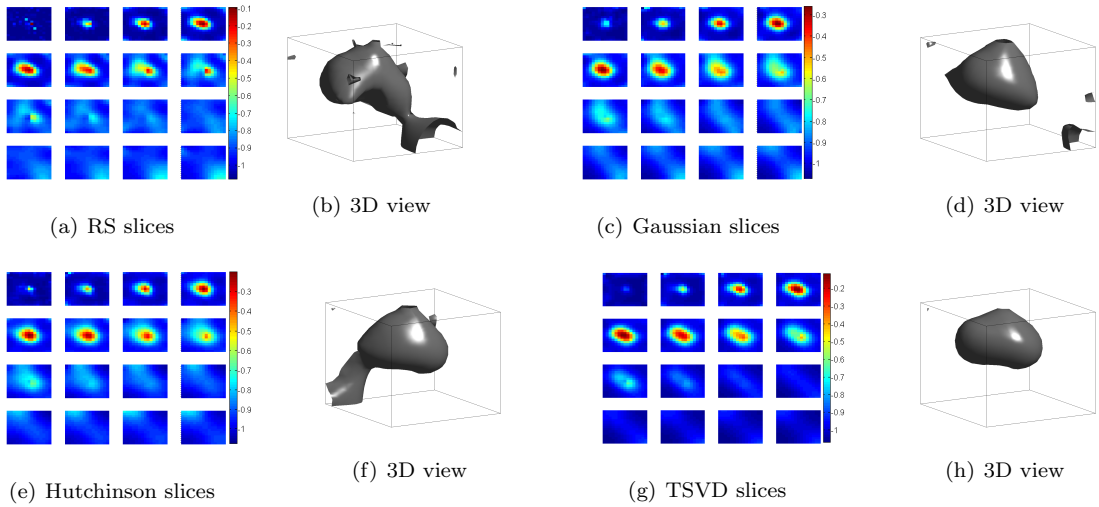


FIG. 5.8. *True Model for Examples 3 and 4. The left panel shows 2D equi-distant slices in the z direction from top to bottom, the right panel depicts the 3D volume.*



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

FIG. 5.9. *Example 3 – reconstructed log conductivity for the 3D model using Algorithm 1 and (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.*

in the last row of Table 5.1 might be deceiving at first glance, as Random Subset seems to be worse than the rest; however, the graph of the misfit in Figure 5.14 reflects a more complete story. At some point in between the final PDE counts for Hutchinson and TSVD, the Random Subset misfit falls below the desired tolerance; however, the uncertainty check at that iterate results in a "false negative" which in turn does not trigger the stopping criterion. This demonstrates the importance of having a very good and reliable trace estimator in the uncertainty check. For all our eight algorithm variants and in all of our examples, we used the Hutchinson trace estimator for this purpose, as it has the smallest variance. And yet, one wrong estimate could result in additional, unnecessary GN iterations, leading to more PDE solves. False positives, on the other hand, trigger an unnecessary stopping criterion evaluation, which results in more PDE solves to calculate the misfit on the entire data set.

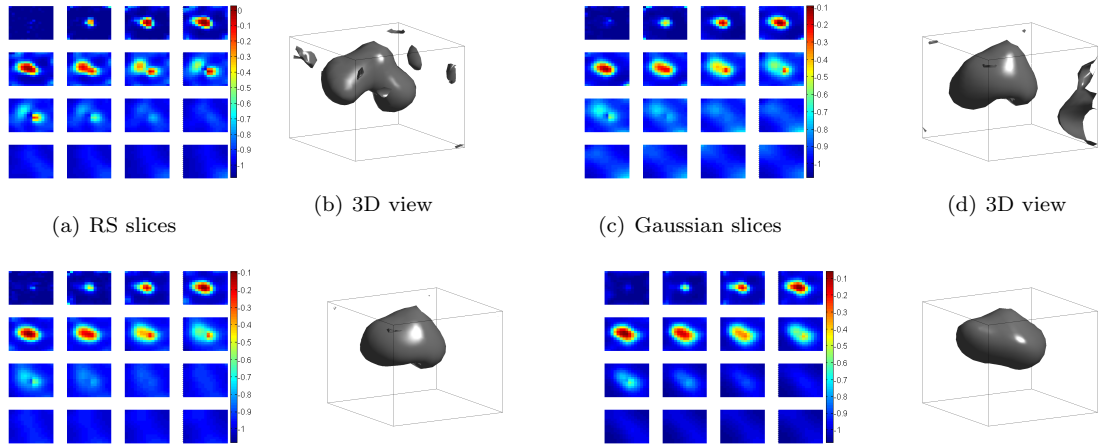
For this example it was also observed that typically the Gaussian method outperforms Hutchinson by a factor of roughly 1.5.

6. Conclusions and future work. In this paper we have developed and compared several highly efficient stochastic algorithms for the solution of inverse problems involving PDEs (specifically, in the context of DC resistivity and EIT problems), in the presence of many measurements or experiments s. Two algorithms for controlling the size $s_n \leq s$ of the data set in the nth stabilized GN iteration have been proposed and tested. For each, four methods for selecting the weight matrix W were proposed, three stochastic and one deterministic, making for a total of eight algorithm variants. Our algorithms are known to converge under suitable circumstances because they satisfy the general conditions in [16, 7].

It is important to emphasize that any of these algorithms is far better than a straightforward utilization of all experiments at each GN iteration. This is clearly borne out in Table 5.1. Note further that in order to facilitate a fair comparison we chose a fixed number of PCG inner iterations, ignoring the adaptive Algorithm 1 of [11], even though that algorithm can impact performance significantly. We also utilized for the sake of fair comparison a rather rigid (and expensive) stopping criterion. Further, we use the Hutchinson estimator for the uncertainty check in all methods, thus making them all stochastic. In particular, TSVD may not be used in (3.2) because it does not lead to a good unbiased estimator for the objective function ϕ when $s_n \ll s$.

Inverse problems with many measurements arise in different applications which may have very different solution sensitivity to changes in the data (e.g., the full waveform inversion, although having other big difficulties in its solution process, is far less sensitive in this sense than DC resistivity). But in any case, it is an accepted working assumption that more data can only help and not hurt the conditioning of the problem being solved. This then gives rise to the question whether our model reduction techniques may worsen the conditioning of the given problem. We have not observed any such effect in our experiments (and our "Vanilla" reconstructions in Section 5 are never better, or sharper, than the other, cheaper ones). In a sense it could be argued that a good model reduction algorithm actually covers approximately the same grounds as the full data problem, so it achieves a similar level of solution sensitivity to data.

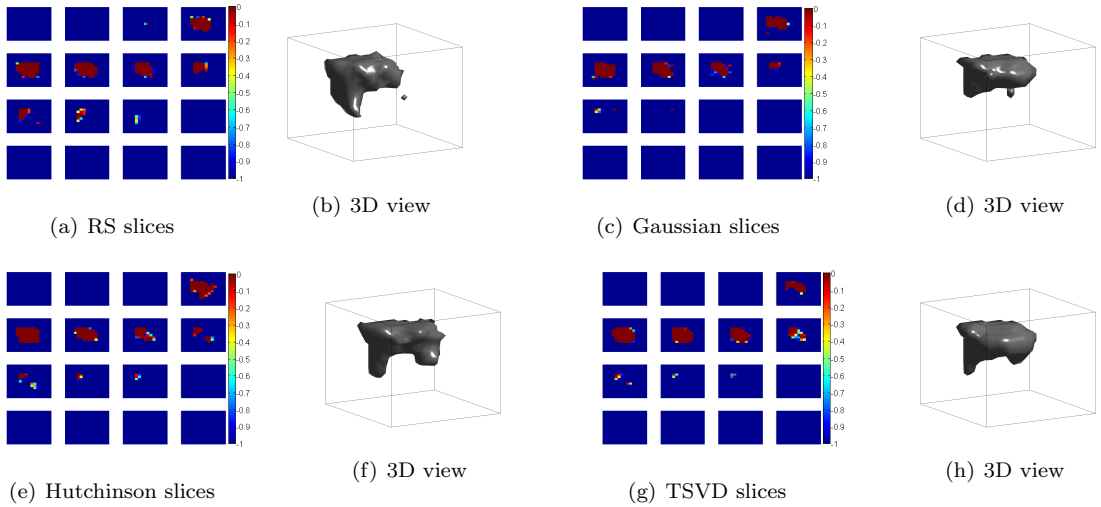
As demonstrated in Examples 2 and 3, neither Algorithm 1 nor Algorithm 2 is always better than the other, and they often both perform well. Their relative performance depends on circumstances that can occasionally be distinguished before committing to calculations. Specifically, if there are relatively few data sets, as in Example 3, then Algorithm 1 is preferable, being both simpler and



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

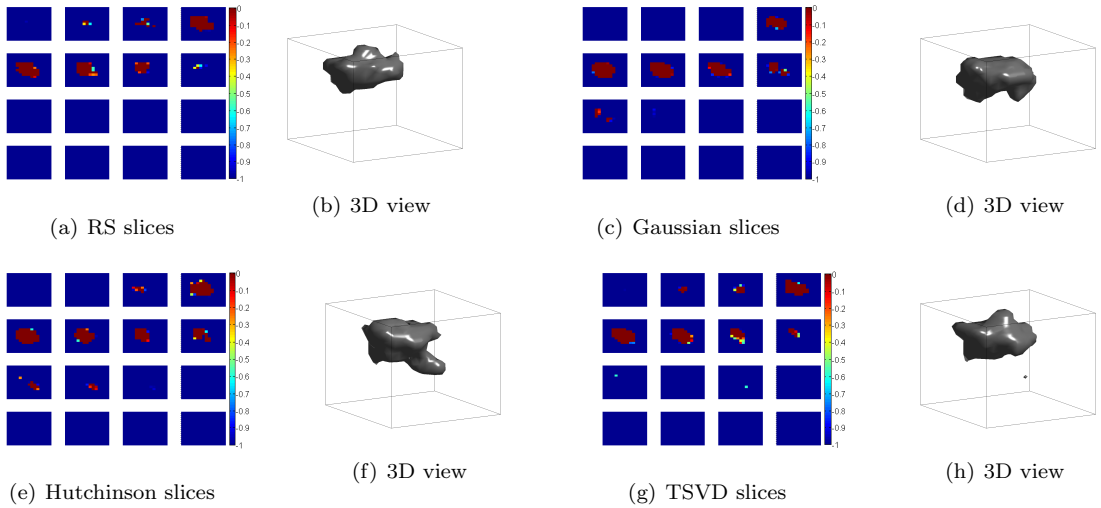
FIG. 5.10. *Example 3 – reconstructed log conductivity for the 3D model using Algorithm 2 and (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.*



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

FIG. 5.11. *Example 4 – reconstructed log conductivity for the 3D model using the level set method with Algorithm 1 and with (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.*



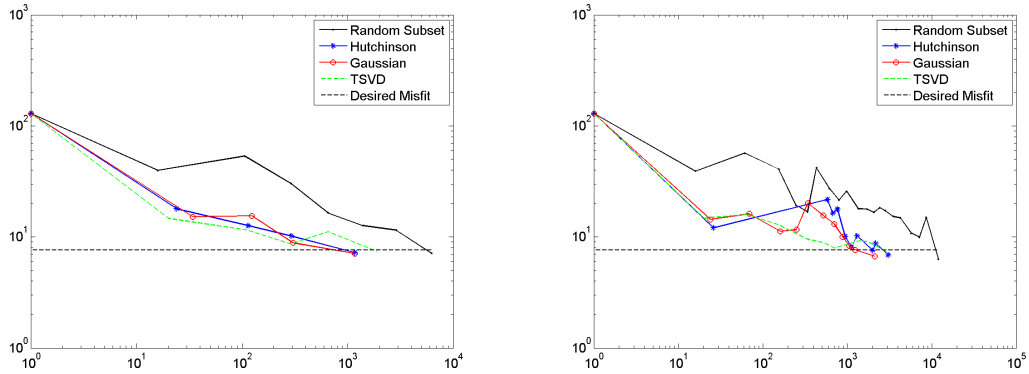
(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

FIG. 5.12. *Example 4 – reconstructed log conductivity for the 3D model using the level set method with Algorithm 2 and with (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.*

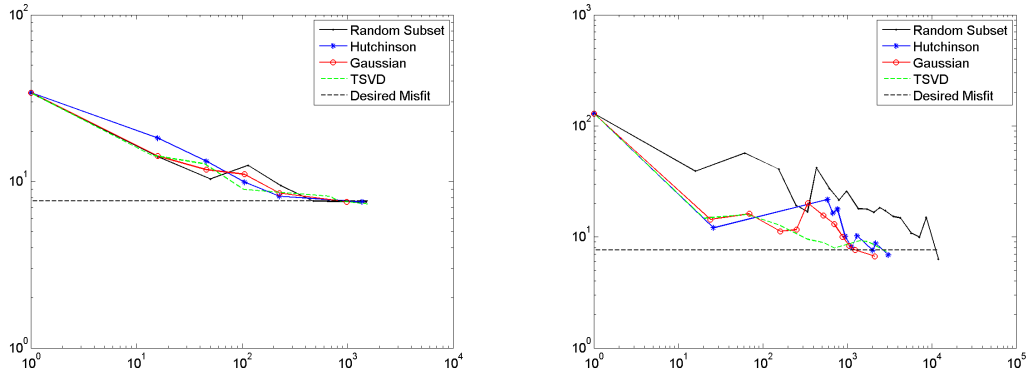
occasionally faster. On the other hand, if s is very large, the data having been massively calculated without much regard to experimental design considerations (as is often the case in geophysical exploration applications), then this may naturally lead to a case of embarrassing redundancy, and caution alone dictates using Algorithm 2.

The three methods of simultaneous sources, namely, Hutchinson, Gaussian and TSVD, are comparable (ignoring the cost of SVD computation), and no definitive answer can be given as to which is better for the model reduction W_n^f. Further, especially when the level set method may not be used, we have found the methods of simultaneous sources to be consistently more efficient than the Random Subset method of [11], roughly by a factor of two. This raises the question whether the restriction that the projection matrices P_i must not depend on i (i.e., shared receivers) can somehow



(a) Algorithm 1 (b) Algorithm 2

FIG. 5.13. *Data misfit vs. PDE count for Example 3.*



(a) Algorithm 1 (b) Algorithm 2

FIG. 5.14. *Data misfit vs. PDE count for Example 4.*

be removed or relaxed, thus allowing use of the faster methods of simultaneous sources. This is a subject of investigation that we plan to carry out in the near future.

Appendix A. Here we describe the forward problem that yields the operators $\mathbf{F}_i(\mathbf{m})$ of (1.4b), and provide some detail on the stabilized GN iteration used in our numerical experiments. There is nothing strictly new here, and yet some of the details are both tricky and very important for the success of an efficient code for computing reasonable reconstructions for this highly ill-posed problem. It is therefore convenient to gather all these details in one place for further reference.

Discretizing the forward problem. The PDE (5.1) is discretized on a staggered grid as described in [3] and in Section 3.1 of [2]. The domain is divided into uniform cells of side length h, and a cell-nodal discretization is employed, where the field unknowns $u_{i,j}$ (or $u_{i,j,k}$ in 3D) are perceived to be located at the cell corners (which are cell centers of the dual grid) while $\sigma_{i+1/2,j+1/2}$ values are at cell centers (cell corners of the dual grid). For the finite volume derivation, the PDE

(5.1a) is written first as

$$\mathbf{j} = \sigma(\mathbf{x}) \operatorname{grad} u, \quad \mathbf{x} \in \Omega, \tag{6.1a}$$

$$\nabla \cdot \mathbf{j} = q(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{6.1b}$$

and then both first order PDEs are integrated prior to discretization. A subsequent, standard removal of the constant null-space then results in the discretized problem (1.4a).

To solve the resulting linear system, since there are a large number of right hand sides in (1.4a), a direct method which involves one Cholesky decomposition followed by forward and backward substitution for each right hand side is highly recommended. If the program runs out of memory (on our system this happens in 3D for $h = 2^{-6}$) then we use a preconditioned conjugate gradient method with an incomplete Cholesky decomposition for a preconditioner.

Taking advantage of additional a priori information. In general, we wish to recover $\sigma(\mathbf{x})$ based on measurements of the field $u(\mathbf{x})$ such that (5.1) approximately holds. Note that, since the measurements are made only at relatively few locations, e.g., the domain's boundary rather than every point in its interior, the matrices P_i (whether or not they are all equal) all have significantly more columns than rows. Moreover, this inverse problem is notoriously ill-posed and difficult in practice, especially for cases where σ has large-magnitude gradients. Below we introduce additional a priori information, when such is available, via a parametrization of $\sigma(\mathbf{x})$ in terms of $m(\mathbf{x})$ (see also [12]). To this end let us define the transfer function

$$\psi(\tau) = \psi(\tau; \theta, \alpha_1, \alpha_2) = \alpha \tanh\left(\frac{\tau}{\alpha\theta}\right) + \frac{\alpha_1 + \alpha_2}{2}, \quad \alpha = \frac{\alpha_2 - \alpha_1}{2}. \tag{6.2}$$

This maps the real line into the interval (α_1, α_2) with the maximum slope θ^{-1} attained at $\tau = 0$.

1. In practice, often there are reasonably tight bounds available, say $\sigma_{\min}$ and $\sigma_{\max}$, such that $\sigma_{\min} \le \sigma(\mathbf{x}) \le \sigma_{\max}$. Such information may be enforced using (6.2) by defining

$$\sigma(\mathbf{x}) = \psi(m(\mathbf{x})), \quad \text{with } \psi(\tau) = \psi(\tau; 1, \sigma_{\min}, \sigma_{\max}). \tag{6.3}$$

2. Occasionally it is reasonable to assume that the sought conductivity function $\sigma(\mathbf{x})$ takes only one of two values, σ_I or σ_{II}, at each $\mathbf{x}$. Viewing one of these as a background value, the problem is that of shape optimization. Such an assumption greatly stabilizes the inverse problem [1]. In [9, 10, 11] we considered an approximate level set function representation for the present problem. We write $\sigma(\mathbf{x}) = \lim_{h\to 0} \sigma(\mathbf{x}; h)$, where

$$\sigma(\mathbf{x}; h) = \psi(m(\mathbf{x}); h, \sigma_I, \sigma_{II}). \tag{6.4}$$

The function $\psi(\tau; h)$ depends on the resolution, or grid width h. It is a scaled and mollified version of the Heaviside step function, and its derivative magnitude is at most $O(\frac{|\sigma_I - \sigma_{II}|}{h})$. Thus, as $h \to 0$ the sought function $m(\mathbf{x})$ satisfying

$$\begin{aligned} \nabla \cdot (\psi(m(\mathbf{x})) \operatorname{grad} \mathbf{u}_i) &= \mathbf{q}_i, \quad i = 1, \ldots, s, \\ \frac{\partial \mathbf{u}_i}{\partial n}\Big|_{\partial\Omega} &= 0, \end{aligned} \tag{6.5}$$

has bounded first derivatives, whereas $\sigma(\mathbf{x})$ is generally discontinuous.

Establishing the relationship between σ and m completes the definition of the forward operators $\mathbf{F}_i(\mathbf{m})$ by (1.4b).

Stabilized Gauss-Newton. Here we briefly describe the modifications made to the GN method (2.5), turning it into the stabilized GN method used in our experiments. The first modification is to replace (2.5b) by

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \gamma \delta \mathbf{m}, \tag{6.6}$$

where γ, $0 < \gamma \leq 1$, is determined by a weak line search ensuring sufficient decrease in $\hat{\phi}(\mathbf{m}_{n+1}, W_n)$ as compared to $\hat{\phi}(\mathbf{m}_n, W_n)$.

More importantly, the matrix at the left hand side of (2.5a) is singular in the usual case where $l < l_m$, and therefore this linear system requires regularization. Furthermore, $\delta\mathbf{m}$ also requires smoothing, because there is nothing in (2.5) to prevent it from forming a non-smooth grid function. These regularization tasks are achieved by applying only a small number of PCG iterations towards the solution of (2.5a), see [10, 11]. This dynamic regularization (or iterative regularization [21]) is also very efficient, and results in a *stabilized GN* iteration. An adaptive algorithm for determining a good number of such inner iterations is proposed in [11]. However, here we opt to keep this number fixed at r PCG iterations independently of n, in order to be able to compare other aspects of our algorithms more fairly. Further, the task of penalizing excessive non-smoothness in the correction $\delta\mathbf{m}$ is achieved by choosing as the preconditioner a discrete Laplacian with homogeneous Neumann boundary conditions. This corresponds to a penalty on $\int |\operatorname{grad} m(\mathbf{x})|^2$ (i.e., least squares but *not* total variation).

The modified GN iteration described above is our outer iteration, and the entire regularization method is called *dynamical regularization* [10, 11]. The essential cost in terms of PDE solves comes through (2.5a) from multiplying J_{ni} or J_{ni}^T by a vector. For the case $P_i = P\ \forall i$, each such multiplication costs one PDE solve, hence $2rs_n$ solves for the left hand side of (2.5a). The evaluation of the gradient costs another $2s_n$ PDE solves per outer iteration. This gives the work under-estimate formula (2.6). This still neglects, for clarity, the additional line search costs, although the additional PDE solves necessitated for determining γ in (6.6) have of course been counted and included in the work tallies reported in Section 5.

REFERENCES

[1] A. Alessandrini and S. Vessella. Lipschitz stability for the inverse conductivity problem. *Adv. Appl. Math.*, 35:207–241, 2005.

[2] U. Ascher. *Numerical Methods for Evolutionary Differential Equations*. SIAM, Philadelphia, PA, 2008.

[3] U. Ascher and E. Haber. A multigrid method for distributed parameter estimation problems. *J. ETNA*, 18:1–18, 2003.

[4] K. Astala and L. Paivarinta. Calderon inverse conductivity problem in the plane. *Annals of Math.*, 163:265–299, 2006.

[5] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *JACM*, 58(2), 2011. Article 8.

[6] L. Borcea, J. G. Berryman, and G. C. Papanicolaou. High-contrast impedance tomography. *Inverse Problems*, 12:835–858, 1996.

[7] R. Byrd, G. Chin, W. Neveitt, and J. Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM J. Optimization*, 21(3):977–995, 2011.

[8] M. Cheney, D. Isaacson, and J. C. Newell. Electrical impedance tomography. *SIAM Review*, 41:85–101, 1999.

[9] K. van den Doel and U. Ascher. On level set regularization for highly ill-posed distributed parameter estimation problems. *J. Comp. Phys.*, 216:707–723, 2006.

[10] K. van den Doel and U. Ascher. Dynamic level set regularization for large distributed parameter estimation problems. *Inverse Problems*, 23:1271–1288, 2007.

[11] K. van den Doel and U. Ascher. Adaptive and stochastic algorithms for EIT and DC resistivity problems with piecewise constant solutions and many measurements. *SIAM J. Scient. Comput.*, 34:DOI: 10.1137/110826692, 2012.

[12] K. van den Doel, U. Ascher, and E. Haber. The lost honour of ℓ_2-based regularization. *Radon Series in Computational and Applied Math*, 2013. M. Cullen, M. Freitag, S. Kindermann and R. Scheinchl (Eds).

[13] O. Dorn, E. L. Miller, and C. M. Rappaport. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, 16, 2000. 1119-1156.

[14] J. Krebs et al. Iterative inversion of data from simultaneous geophysical sources. *http://www.faqs.org/patents/app/20100018718*, 2010.

[15] A. Fichtner. *Full Seismic Waveform Modeling and Inversion*. Springer, 2011.

[16] M. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Scient. Comput.*, 34(3), 2012.

[17] S. Geisser. *Predictive Inference*. New York: Chapman and Hall, 1993.

[18] E. Haber, U. Ascher, and D. Oldenburg. Inversion of 3D electromagnetic data in frequency and time domain

using an inexact all-at-once approach. *Geophysics*, 69:1216–1228, 2004.

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

[20] E. Haber, S. Heldmann, and U. Ascher. Adaptive finite volume method for distributed non-smooth parameter identification. *Inverse Problems*, 23:1659–1676, 2007.

[21] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, 1998.

[22] F. Herrmann, Y. Erlangga, and T. Lin. Compressive simultaneous full-waveform simulation. *Geophysics*, 74:A35, 2009.

[23] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *J. Comm. Stat. Simul.*, 19:433–450, 1990.

[24] A. Juditsky, G. Lan, A. Nemirovski, and A. Shapiro. Stochastic approximation approach to stochastic programming. *SIAM J. Optimization*, 19(4):1574–1609, 2009.

[25] G. A. Newman and D. L. Alumbaugh. Frequency-domain modelling of airborne electromagnetic responses using staggered finite differences. *Geophys. Prospecting*, 43:1021–1042, 1995.

[26] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.

[27] D. Oldenburg, E. Haber, and R. Shekhtman. 3D inverseion of multi-source time domain electromagnetic data. *J. Geophysics*, 2013. To appear.

[28] L. Paivarinta, A. Panchenko, and G. Uhlmann. Complex geometrical optics solutions for Lipschitz conductivities. *Rev. Mat. Iberoamericana*, 19:57–72, 2003.

[29] A. Pidlisecky, E. Haber, and R. Knight. RESINVM3D: A MATLAB 3D Resistivity Inversion Package. *Geophysics*, 72(2):H1–H10, 2007.

[30] J. Rohmberg, R. Neelamani, C. Krohn, J. Krebs, M. Deffenbaugh, and J. Anderson. Efficient seismic forward modeling and acquisition using simultaneous random sources and sparsity. *Geophysics*, 75(6):WB15–WB27, 2010.

[31] A. Shapiro, D. Dentcheva, and D. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. Piladelphia: SIAM, 2009.

[32] N. C. Smith and K. Vozoff. Two dimensional DC resistivity inversion for dipole dipole data. *IEEE Trans. on geoscience and remote sensing*, GE 22:21–28, 1984.

[33] C. Vogel. *Computational methods for inverse problem*. SIAM, Philadelphia, 2002.